



# Fast Models

Version 11.29

## Reference Guide

**Non-Confidential**

Copyright © 2017–2025 Arm Limited (or its affiliates).  
All rights reserved.

**Issue 00**

100964\_1129\_00\_en



## Fast Models Reference Guide

This document is Non-Confidential.

Copyright © 2017–2025 Arm Limited (or its affiliates). All rights reserved.

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

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

This document (100964\_1129\_00\_en) was issued on 2025-05-16. There might be a later issue at <https://developer.arm.com/documentation/100964>

The product version is 11.29.

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

### Start reading

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

### Intended audience

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

### Inclusive language commitment

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

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

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

### Feedback

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

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

# Contents

<b>1. About the models.....</b>	<b>21</b>
1.1 Model capabilities.....	21
1.2 Running Arm Software Test Libraries (STL) on Fast Models.....	21
1.3 Quality level definitions.....	22
1.4 Fast Models accuracy.....	22
1.4.1 Timing accuracy of Fast Models.....	23
1.4.2 Bus traffic in Fast Models.....	23
1.4.3 Instruction prefetch in Fast Models.....	25
1.4.4 Out-of-order execution and write-buffers in Fast Models.....	25
1.4.5 Caches in Fast Models.....	26
1.4.6 Global exclusive monitor in Fast Models.....	26
1.5 Processor implementation.....	27
1.5.1 Mapping PMU events to MTI trace sources.....	27
1.5.2 Caches in PV models.....	29
1.5.3 GICv3 in PV models.....	31
1.5.4 GICv4 in PV models.....	31
1.5.5 CP14 Debug coprocessor.....	32
1.5.6 TLBs in PV models.....	32
1.5.7 Memory access in PV models.....	32
1.5.8 Timing in PV models.....	33
1.6 syncLevel definitions.....	34
1.7 Controlling and observing the syncLevel.....	35
1.8 User mode networking.....	37
1.9 TAP/TUN networking.....	38
1.9.1 TAP/TUN networking limitations.....	38
1.9.2 Setting up a network connection for Red Hat Enterprise Linux.....	38
1.9.3 Setting up a network connection for Ubuntu Linux.....	40
1.9.4 Configuring the networking environment for Linux.....	42
1.9.5 Solutions to networking issues on Linux.....	43
1.9.6 Disabling and re-enabling networking for Linux.....	43
1.9.7 Uninstalling networking for Linux.....	44
1.10 Using parameters to set port values.....	44

1.11 PVBUS C++ transaction and Tx_Result classes.....	45
1.11.1 Class pv::TransactionGenerator.....	45
1.11.2 TransactionGenerator efficiency considerations.....	46
1.11.3 Enum pv::AccessWidth.....	46
1.11.4 Class pv::Transaction.....	46
1.11.5 Class pv::ReadTransaction.....	47
1.11.6 Class pv::WriteTransaction.....	47
1.12 Visualisation library.....	48
1.12.1 LISA visualisation models.....	49
1.12.2 Visualisation library C++ classes.....	49
<b>2. Protocols.....</b>	<b>54</b>
2.1 AMBAPV protocol.....	54
2.2 AMBAPVACE protocol.....	57
2.3 AMBAPVSignal protocol.....	60
2.4 AMBAPVSignalState protocol.....	60
2.5 AMBAPVValue protocol.....	61
2.6 AMBAPVValue64 protocol.....	61
2.7 AMBAPVValueState protocol.....	61
2.8 AMBAPVValueState64 protocol.....	62
2.9 AsyncSignalCallback protocol.....	63
2.10 AsyncSignalControl protocol.....	63
2.11 AudioControl protocol.....	63
2.12 CADIDisassemblerProtocol protocol.....	64
2.13 CADIProtocol protocol.....	66
2.14 CCI500_AddressDecoderProtocol protocol.....	69
2.15 CCIInterconnectControl protocol.....	71
2.16 ClockRateControl protocol.....	71
2.17 ClockSignal protocol.....	72
2.18 CompoundPortLisa protocol.....	73
2.19 CoprocBusProtocol protocol.....	74
2.20 CounterInterface protocol.....	78
2.21 DVMMMessage protocol.....	79
2.22 EventBus protocol.....	79
2.23 Feature protocol.....	79
2.24 FlashLoaderPort protocol.....	80



2.25 FrameTracingProtocol protocol.....	80
2.26 GICv3Comms protocol.....	81
2.27 GUIPollCallback protocol.....	81
2.28 ICS307Configuration protocol.....	82
2.29 InstructionCount protocol.....	82
2.30 KeyboardStatus protocol.....	83
2.31 LCD protocol.....	84
2.32 LCDLayoutInfo protocol.....	85
2.33 MMC_Protocol protocol.....	85
2.34 MMU_400_BASE_IDENTIFY protocol.....	87
2.35 MMU_400_Internals protocol.....	87
2.36 MMU_500_BASE_IDENTIFY protocol.....	88
2.37 MMU_500_Internals protocol.....	88
2.38 MouseStatus protocol.....	89
2.39 PASSwitchControl protocol.....	89
2.40 PCIDevice2ClientProtocol protocol.....	91
2.41 PCIeATC_get_if protocol.....	92
2.42 PChannel protocol.....	93
2.43 PL080_DMACE_DmaPortProtocol protocol.....	94
2.44 PL330CcppToLISA protocol.....	95
2.45 PL330_DMACE_DmaPortProtocol protocol.....	95
2.46 PMUEvent protocol.....	96
2.47 PS2Data protocol.....	96
2.48 PVBus protocol.....	97
2.49 PVBus2PCI2PCIDeviceProtocol protocol.....	100
2.50 PVBusBridgeControl protocol.....	101
2.51 PVBusCacheControl protocol.....	102
2.52 PVBusCacheDevice protocol.....	104
2.53 PVBusMapperControl protocol.....	105
2.54 PVBusOverTLMControl protocol.....	108
2.55 PVBusRouterControl protocol.....	109
2.56 PVBusSlaveControl protocol.....	109
2.57 PVC2C protocol.....	114
2.58 PVCacheDebugRam protocol.....	115
2.59 PVCacheMaintenance protocol.....	117
2.60 PVDevice protocol.....	122

2.61 PVTransactionMaster protocol.....	123
2.62 PVWriteBuffer_BarrierPort protocol.....	124
2.63 PVWriteBuffer_SErrorPort protocol.....	125
2.64 PVWriteBuffer_VmidBarrierPort protocol.....	125
2.65 SC_ClockRateControl protocol.....	125
2.66 SC_ClockSignal protocol.....	126
2.67 SC_VirtualEthernet protocol.....	128
2.68 SMMUv3AEMIdentifyProtocol protocol.....	129
2.69 SchedulerInterfaceControl protocol.....	130
2.70 SchedulerThreadControl protocol.....	130
2.71 SchedulerThreadEventControl protocol.....	131
2.72 SerialData protocol.....	132
2.73 Signal protocol.....	133
2.74 StateSignal protocol.....	134
2.75 SystemCCoprocBusProtocol protocol.....	134
2.76 SystemCPChannel protocol.....	135
2.77 SystemCoherencyInterface protocol.....	135
2.78 TZFilterControl protocol.....	136
2.79 TZSwitchControl protocol.....	137
2.80 TimerCallback protocol.....	138
2.81 TimerCallback64 protocol.....	138
2.82 TimerControl protocol.....	139
2.83 TimerControl64 protocol.....	140
2.84 VECBProtocol protocol.....	140
2.85 VGICComponentTraceExport protocol.....	141
2.86 VGICReportingProtocol protocol.....	141
2.87 Value protocol.....	142
2.88 ValueState protocol.....	142
2.89 ValueState_64 protocol.....	143
2.90 Value_64 protocol.....	143
2.91 VirtualEthernet protocol.....	144
2.92 VisEventRecorderProtocol protocol.....	144
2.93 v7_VGIC_Configuration_Protocol protocol.....	145
2.94 v8EmbeddedCrossTrigger_controlprotocol protocol.....	146

### **3. Fast Models components..... 148**

3.1 Component differences.....	148
3.2 Bridge components.....	149
3.2.1 AMBAPV2PVBUS.....	150
3.2.2 AMBAPVACE2PVBUS.....	152
3.2.3 AMBAPVSignal2SGSignal.....	153
3.2.4 AMBAPVSignalState2SGStateSignal.....	153
3.2.5 AMBAPVValue2SGValue.....	154
3.2.6 AMBAPVValue2SGValue64.....	154
3.2.7 AMBAPVValue642SMMUv3AEMIdentify.....	155
3.2.8 AMBAPVValue642VECB.....	155
3.2.9 AMBAPVValueState2SGValueState.....	156
3.2.10 AMBAPVValueState2SGValueState64.....	156
3.2.11 BroadcastSignal2AMBAPVSignal.....	157
3.2.12 Clock2SystemC.....	157
3.2.13 ClockRateConversion.....	158
3.2.14 ClockSignal2SC_ClockSignal.....	160
3.2.15 CoprocBus2SystemC.....	161
3.2.16 CounterInterface2SystemC.....	161
3.2.17 InstructionCount2SystemC.....	162
3.2.18 LCD2SystemC.....	162
3.2.19 PChannel2SystemC.....	163
3.2.20 PVBUS2AMBAPV.....	164
3.2.21 PVBUS2AMBAPVACE.....	166
3.2.22 PVBUSBridge.....	170
3.2.23 SC_ClockSignal2ClockSignal.....	170
3.2.24 SGSignal2AMBAPVSignal.....	171
3.2.25 SGStateSignal2AMBAPVSignalState.....	171
3.2.26 SGValue2AMBAPVValue.....	172
3.2.27 SGValue2AMBAPVValue64.....	173
3.2.28 SGValueState2AMBAPVValueState.....	173
3.2.29 SGValueState2AMBAPVValueState64.....	174
3.2.30 SMMUv3AEMIdentify2AMBAPVValue64.....	175
3.2.31 SystemC2Clock.....	175
3.2.32 SystemC2CoprocBus.....	176
3.2.33 SystemC2CounterInterface.....	176
3.2.34 SystemC2InstructionCount.....	177

3.2.35 SystemC2LCD.....	177
3.2.36 SystemC2PChannel.....	178
3.2.37 SystemC2VirtualEthernet.....	178
3.2.38 SystemC2v7VGICConfig.....	179
3.2.39 VECB2AMBAPVValue64.....	179
3.2.40 VirtualEthernet2SystemC.....	180
3.2.41 v7VGICConfig2SystemC.....	180
3.3 Bus components.....	181
3.3.1 Labeller.....	182
3.3.2 LabellerForDMA330.....	184
3.3.3 LabellerForGPUProtMode.....	186
3.3.4 MSIRewriter.....	187
3.3.5 PASSwitch.....	191
3.3.6 PVBUS4KBTo1KBSplitter.....	194
3.3.7 PVBUSCache.....	195
3.3.8 PVBUSDecoder.....	196
3.3.9 PVBUSExclusiveMonitor.....	197
3.3.10 PVBUSExclusiveSquasher.....	200
3.3.11 PVBUSLogger.....	200
3.3.12 PVBUSMapper.....	202
3.3.13 PVBUSMaster.....	203
3.3.14 PVBUSModifier.....	204
3.3.15 PVBUSRouter.....	205
3.3.16 PVBUSSlave.....	206
3.3.17 PVCoherentInterconnect.....	208
3.3.18 PVMemoryProtectionEngine.....	220
3.3.19 PVWriteBuffer.....	223
3.3.20 SimplePVBUSMaster.....	224
3.3.21 TZSwitch.....	225
3.4 Clocking components.....	227
3.4.1 ClockDivider.....	227
3.4.2 ClockGate.....	229
3.4.3 ClockSelector.....	230
3.4.4 ClockTimer.....	236
3.4.5 ClockTimer64.....	237
3.4.6 ClockTimerThread.....	238

3.4.7 ClockTimerThread64.....	238
3.4.8 ClusterClockControl.....	239
3.4.9 MasterClock.....	247
3.4.10 PLLControl.....	248
3.4.11 ScalableClockControl.....	249
3.4.12 SwitchedClockControl.....	261
3.5 Core components.....	269
3.5.1 AEMv8RMPCT.....	270
3.5.2 AEMvACT.....	429
3.5.3 ARMAEMv8MCT.....	699
3.5.4 ARMCortexA5CT.....	976
3.5.5 ARMCortexA5MPx1CT.....	989
3.5.6 ARMCortexA7x1CT.....	1004
3.5.7 ARMCortexA8CT.....	1023
3.5.8 ARMCortexA9MPx1CT.....	1037
3.5.9 ARMCortexA9UPCT.....	1053
3.5.10 ARMCortexA15x1CT.....	1066
3.5.11 ARMCortexA17x1CT.....	1085
3.5.12 ARMCortexA32CT.....	1104
3.5.13 ARMCortexA34CT.....	1126
3.5.14 ARMCortexA35CT.....	1147
3.5.15 ARMCortexA53CT.....	1169
3.5.16 ARMCortexA55CT.....	1193
3.5.17 ARMCortexA55CT_CortexA75CT.....	1218
3.5.18 ARMCortexA55CT_CortexA76CT.....	1258
3.5.19 ARMCortexA55CT_CortexA78CT.....	1298
3.5.20 ARMCortexA57CT.....	1339
3.5.21 ARMCortexA65AECT.....	1361
3.5.22 ARMCortexA65AECT_CortexA76AECT.....	1389
3.5.23 ARMCortexA65CT.....	1432
3.5.24 ARMCortexA72CT.....	1458
3.5.25 ARMCortexA73CT.....	1479
3.5.26 ARMCortexA75CT.....	1501
3.5.27 ARMCortexA76AECT.....	1527
3.5.28 ARMCortexA76CT.....	1554
3.5.29 ARMCortexA77CT.....	1580

3.5.30 ARMCortexA78AECT.....	1606
3.5.31 ARMCortexA78CCT.....	1634
3.5.32 ARMCortexA78CT.....	1660
3.5.33 ARMCortexA320CT.....	1686
3.5.34 ARMCortexA510CT.....	1718
3.5.35 ARMCortexA510CT_CortexA710CT.....	1751
3.5.36 ARMCortexA510CT_CortexA710CT_CortexX2CT.....	1804
3.5.37 ARMCortexA510CT_CortexA710CT_CortexX3CT.....	1877
3.5.38 ARMCortexA510CT_CortexA715CT_CortexX3CT.....	1952
3.5.39 ARMCortexA520AECT.....	2029
3.5.40 ARMCortexA520CT.....	2065
3.5.41 ARMCortexA520CT_CortexA720CT.....	2100
3.5.42 ARMCortexA520CT_CortexA720CT_CortexX4CT.....	2156
3.5.43 ARMCortexA520CT_CortexA725CT.....	2234
3.5.44 ARMCortexA520CT_CortexA725CT_CortexX925CT.....	2290
3.5.45 ARMCortexA710CT.....	2367
3.5.46 ARMCortexA715CT.....	2400
3.5.47 ARMCortexA720AECT.....	2435
3.5.48 ARMCortexA720CT.....	2470
3.5.49 ARMCortexA725CT.....	2505
3.5.50 ARMCortexA725CT_CortexX925CT.....	2539
3.5.51 ARMCortexM0CT.....	2595
3.5.52 ARMCortexM0PlusCT.....	2601
3.5.53 ARMCortexM3CT.....	2608
3.5.54 ARMCortexM4CT.....	2615
3.5.55 ARMCortexM7CT.....	2623
3.5.56 ARMCortexM23CT.....	2635
3.5.57 ARMCortexM33CT.....	2653
3.5.58 ARMCortexM35PCT.....	2675
3.5.59 ARMCortexM52CT.....	2696
3.5.60 ARMCortexM55CT.....	2719
3.5.61 ARMCortexM85CT.....	2742
3.5.62 ARMCortexR4CT.....	2765
3.5.63 ARMCortexR5x1CT.....	2777
3.5.64 ARMCortexR7x1CT.....	2791
3.5.65 ARMCortexR8x1CT.....	2806

3.5.66 ARMCortexR52CT.....	2822
3.5.67 ARMCortexR52PlusCT.....	2843
3.5.68 ARMCortexR82AECT.....	2861
3.5.69 ARMCortexR82CT.....	2892
3.5.70 ARMCortexX1CCT.....	2923
3.5.71 ARMCortexX1CT.....	2949
3.5.72 ARMCortexX2CT.....	2976
3.5.73 ARMCortexX3CT.....	3006
3.5.74 ARMCortexX4CT.....	3042
3.5.75 ARMCortexX925CT.....	3076
3.5.76 ARMNeoverseE1CT.....	3109
3.5.77 ARMNeoverseN1CT.....	3135
3.5.78 ARMNeoverseN2CT.....	3161
3.5.79 ARMNeoverseN3CT.....	3193
3.5.80 ARMNeoverseV1CT.....	3224
3.5.81 ARMNeoverseV2CT.....	3251
3.5.82 ARMNeoverseV3AECT.....	3284
3.5.83 ARMNeoverseV3CT.....	3319
3.5.84 ARMSC000CT.....	3355
3.5.85 ARMSC300CT.....	3362
3.6 Media components.....	3369
3.6.1 D71.....	3369
3.6.2 FrameTracingComponent.....	3378
3.6.3 Mali_C5x_streaming_sink.....	3382
3.6.4 Mali_C7x_streaming_sink.....	3384
3.6.5 Mali_C55.....	3385
3.6.6 Mali_C71.....	3389
3.6.7 Mali_C78.....	3391
3.6.8 Mali_Cxx_streaming_camera.....	3393
3.6.9 Mali_G71.....	3394
3.6.10 Mali_G76.....	3396
3.6.11 Mali_G78AE.....	3399
3.6.12 Mali_G710.....	3402
3.6.13 Mali_G715.....	3405
3.6.14 Mali_G720.....	3408
3.6.15 Mali_G725.....	3411

3.6.16 Mali_T624.....	3414
3.6.17 V61.....	3415
3.7 Peripheral components.....	3419
3.7.1 Ashbrook_SoC_SCC.....	3419
3.7.2 AudioOut_File.....	3424
3.7.3 AudioOut_SDL.....	3425
3.7.4 Base_PowerController.....	3425
3.7.5 CMSDK_Timer.....	3428
3.7.6 Clock_Multiplexer.....	3430
3.7.7 Cluster_Temperature_Sensor.....	3430
3.7.8 CombinedMessagingUnit.....	3432
3.7.9 CombinedMessagingUnitAE.....	3444
3.7.10 DebugAccessPort.....	3454
3.7.11 DebugROM.....	3456
3.7.12 DualClusterSystemConfigurationBlock.....	3471
3.7.13 DummyAPB.....	3475
3.7.14 ElfLoader.....	3478
3.7.15 FlashLoader.....	3482
3.7.16 GICv3CommsLogger.....	3484
3.7.17 GICv3CommsPVBUS.....	3485
3.7.18 GICv3ProtocolChecker.....	3485
3.7.19 GUIPoll.....	3486
3.7.20 HostBridge.....	3487
3.7.21 HostSerialInterface.....	3489
3.7.22 IntelStrataFlashJ3.....	3490
3.7.23 Interrupt_Router.....	3494
3.7.24 IoTSS3_ManagerSecurityController.....	3496
3.7.25 IoTSS3_MemoryProtectionController.....	3497
3.7.26 IoTSS3_SecureAccessConfig.....	3500
3.7.27 Juno_sysregs.....	3504
3.7.28 Kits2_Timer.....	3505
3.7.29 LS64TestingFIFO.....	3507
3.7.30 LabellerIdauSecurity.....	3508
3.7.31 MemoryMappedGenericTimer.....	3509
3.7.32 MemoryMappedGenericWatchdog.....	3512
3.7.33 NonVolatileCounter.....	3513



3.7.34 PCIeATC.....	3515
3.7.35 PLLClockControl.....	3516
3.7.36 PPUMultiThreadModeSwitch.....	3518
3.7.37 PS2Keyboard.....	3518
3.7.38 PS2Mouse.....	3519
3.7.39 PVBUSGICv3Comms.....	3519
3.7.40 PVMetaDataController.....	3520
3.7.41 PchannelListener.....	3524
3.7.42 PowerStateGate.....	3525
3.7.43 RAMDevice.....	3526
3.7.44 ROM.....	3528
3.7.45 RSE_Integ_Regs.....	3529
3.7.46 RandomNumberGenerator.....	3531
3.7.47 RealTimeLimiter.....	3532
3.7.48 RealtimeClockTimer.....	3533
3.7.49 RemapDecoder.....	3533
3.7.50 RootKeyStorage.....	3534
3.7.51 SI_System_Ctrl_Regs.....	3537
3.7.52 SMCF.....	3540
3.7.53 SMMUv3TestEngine.....	3548
3.7.54 STLBusGasket.....	3550
3.7.55 SerialCrossover.....	3551
3.7.56 Signal_Multiplexer.....	3551
3.7.57 SoC_SOR.....	3552
3.7.58 SoC_Temperature.....	3554
3.7.59 SystemIdUnit.....	3555
3.7.60 TC25_SecureAccessConfig.....	3557
3.7.61 TRNG.....	3561
3.7.62 TelnetTerminal.....	3562
3.7.63 Temperature.....	3567
3.7.64 TestbedGPIOConnector.....	3568
3.7.65 UnusedPrimeCell.....	3569
3.7.66 Value_Multiplexer.....	3569
3.7.67 VirtioBlockDevice.....	3570
3.7.68 VirtioBlockDeviceMMIO.....	3572
3.7.69 VirtioNetMMIO.....	3574

3.7.70 VirtioP9Device.....	3578
3.7.71 VirtioPCIBlockDevice.....	3582
3.7.72 VirtioRNG.....	3584
3.7.73 VirtualEthernetCrossover.....	3587
3.7.74 VirtualEthernetHub3.....	3587
3.7.75 VisEventRecorder.....	3588
3.7.76 Visualisation_sdl2.....	3591
3.7.77 WarningMemory.....	3603
3.7.78 v8EmbeddedCrossTrigger_Matrix.....	3605
3.8 Scheduler components.....	3606
3.8.1 AsyncSignal.....	3606
3.8.2 SchedulerInterface.....	3607
3.8.3 SchedulerThread.....	3607
3.8.4 SchedulerThreadEvent.....	3608
3.9 Signals components.....	3608
3.9.1 AndGate.....	3608
3.9.2 FrequencyProbe.....	3609
3.9.3 LabellerMasterIdExtendedIdUserFlag.....	3609
3.9.4 LabellerUserSignals.....	3612
3.9.5 OrGate.....	3614
3.9.6 SGSignalBuffer.....	3614
3.9.7 SignalDriver.....	3615
3.9.8 SignalInverter.....	3616
3.9.9 SignalLogger.....	3616
3.9.10 Value64Logger.....	3617
3.9.11 ValueLogger.....	3617
3.9.12 WideAndGate.....	3618
3.9.13 WideOrGate.....	3619
3.9.14 WideOrGate_12x4.....	3619
3.10 SystemIP components.....	3620
3.10.1 AHCI_SATA.....	3621
3.10.2 AddressTranslationUnit.....	3623
3.10.3 BP141_TZMA.....	3625
3.10.4 BP147_TZPC.....	3627
3.10.5 CCI400.....	3628
3.10.6 CCI500.....	3633

3.10.7 CCI550.....	3642
3.10.8 CCN502.....	3652
3.10.9 CCN504.....	3661
3.10.10 CCN508.....	3670
3.10.11 CCN512.....	3678
3.10.12 CHBCR.....	3687
3.10.13 CI700.....	3688
3.10.14 CMN600.....	3723
3.10.15 CMN600AE.....	3735
3.10.16 CMN600CMLHub.....	3748
3.10.17 CMN650.....	3750
3.10.18 CMN650R2.....	3764
3.10.19 CMN700.....	3778
3.10.20 CMN_S3.....	3821
3.10.21 DCSU.....	3864
3.10.22 DMA350.....	3865
3.10.23 DMC500.....	3875
3.10.24 DMC520.....	3878
3.10.25 DMC620.....	3880
3.10.26 DMC_400.....	3882
3.10.27 EthosU55.....	3885
3.10.28 EthosU65.....	3887
3.10.29 EthosU85.....	3888
3.10.30 Firewall.....	3891
3.10.31 GIC500.....	3902
3.10.32 GIC500_ClusterPorts.....	3932
3.10.33 GIC500_Filter.....	3962
3.10.34 GIC600.....	3991
3.10.35 GIC600AE.....	4008
3.10.36 GIC600AE_Filter.....	4025
3.10.37 GIC600_Filter.....	4042
3.10.38 GIC625.....	4058
3.10.39 GIC625_Filter.....	4065
3.10.40 GIC700.....	4073
3.10.41 GIC700_Filter.....	4104
3.10.42 GIC720AE.....	4135

3.10.43 GIC720AE_Filter.....	4166
3.10.44 GIC_400.....	4196
3.10.45 GIC_IRI.....	4205
3.10.46 GIC_IRI_Filter.....	4266
3.10.47 ICS307.....	4326
3.10.48 IDAU.....	4329
3.10.49 ILCU.....	4605
3.10.50 IntegrityChecker.....	4606
3.10.51 IoTSS3_SystemControl.....	4608
3.10.52 IoTSS_AccessControlGate.....	4612
3.10.53 IoTSS_PeripheralProtectionController.....	4613
3.10.54 KeyManagementUnit.....	4616
3.10.55 LifeCycleManager.....	4621
3.10.56 MHU320AE.....	4628
3.10.57 MMC.....	4633
3.10.58 MMU_400.....	4640
3.10.59 MMU_400_BASE.....	4671
3.10.60 MMU_500.....	4677
3.10.61 MMU_500_BASE.....	4713
3.10.62 MMU_600.....	4724
3.10.63 MMU_700.....	4753
3.10.64 MMU_720AE.....	4788
3.10.65 MMU_S3.....	4831
3.10.66 MemoryMappedCounterModule.....	4873
3.10.67 MessageHandlingUnit.....	4877
3.10.68 MessageHandlingUnitV2.....	4883
3.10.69 MessageHandlingUnitV3.....	4885
3.10.70 NI700.....	4890
3.10.71 NI710AE.....	4896
3.10.72 NOC_S3.....	4900
3.10.73 OTPW.....	4908
3.10.74 PL011_Uart.....	4909
3.10.75 PL022_SSP.....	4914
3.10.76 PL030_RTC.....	4916
3.10.77 PL031_RTC.....	4917
3.10.78 PL041_AACI.....	4918

3.10.79	PL050_KMI.....	4920
3.10.80	PL061_GPIO.....	4921
3.10.81	PL080_DMAC.....	4922
3.10.82	PL110_CLCD.....	4925
3.10.83	PL111_CLCD.....	4927
3.10.84	PL180_MCI.....	4928
3.10.85	PL192_VIC.....	4930
3.10.86	PL310_L2CC.....	4933
3.10.87	PL330_DMAC.....	4938
3.10.88	PL340_DMC.....	4951
3.10.89	PL350_SMC.....	4961
3.10.90	PL350_SMC_NAND_FLASH.....	4985
3.10.91	PL370_HDLCD.....	4988
3.10.92	PL390_GIC.....	4990
3.10.93	PMU.....	5001
3.10.94	PPUMTWakerequest.....	5006
3.10.95	PPUv0.....	5007
3.10.96	PPUv1.....	5011
3.10.97	PPUv1_Cluster_Wakerequest_Logic.....	5018
3.10.98	RSE_CPU_Private_Region.....	5020
3.10.99	RSE_SystemControl.....	5021
3.10.100	SMMUv3AEM.....	5027
3.10.101	SMSC_91C111.....	5095
3.10.102	SP804_Timer.....	5097
3.10.103	SP805_Watchdog.....	5099
3.10.104	SP810_SysCtrl.....	5100
3.10.105	SSU.....	5103
3.10.106	SecureAlarmManager.....	5105
3.10.107	SecureICache.....	5106
3.10.108	SystemFMU.....	5109
3.10.109	System_RAS_Agent.....	5111
3.10.110	TZC_400.....	5113
3.10.111	TZFilterUnit.....	5126
3.10.112	TZIC.....	5127
3.10.113	TrustedRAM.....	5128
3.10.114	VHT_VIOBridge.....	5129

3.10.115 VHT_VSIBridge.....	5130
3.10.116 VHT_VSocket.....	5131
3.10.117 v7_VGIC.....	5132
<b>4. Plug-ins for Fast Models.....</b>	<b>5136</b>
4.1 Loading a plug-in.....	5136
4.1.1 --plugin command-line option.....	5136
4.1.2 scx::scx_load_plugin() method.....	5136
4.1.3 FM_TRACE_PLUGINS environment variable.....	5137
4.2 Customizing a plug-in.....	5138
4.3 ArchMsgTrace.....	5138
4.3.1 ArchMsgTrace - parameters.....	5140
4.4 ASTFplugin.....	5140
4.4.1 ASTFplugin - parameters.....	5141
4.4.2 ASTFplugin usage notes.....	5142
4.4.3 Additional ASTF support in Fast Models.....	5142
4.4.4 ASTF tools.....	5143
4.4.5 ASTFplugin FAQs.....	5143
4.5 BranchPrediction.....	5145
4.5.1 BranchPrediction - parameters.....	5146
4.5.2 BranchPrediction output example.....	5146
4.5.3 Other ways to report branch mispredictions.....	5147
4.6 CADIIPCRemoteConnection.....	5147
4.7 CDE.....	5148
4.7.1 CDETester.....	5149
4.7.2 CDETester - parameters.....	5150
4.7.3 CDEConstant.....	5151
4.7.4 CDEConstant - parameters.....	5151
4.7.5 Implementing a CDE plug-in.....	5151
4.7.6 CDE API.....	5154
4.8 CDELoader.....	5161
4.8.1 CDELoader - parameters.....	5163
4.8.2 ACI library implementation.....	5163
4.8.3 ACIConstant example ACI library.....	5164
4.8.4 Custom instruction mnemonics.....	5165
4.8.5 ACI library API.....	5165

4.9 Crypto.....	5180
4.9.1 Crypto - parameters.....	5182
4.10 GDBRemoteConnection.....	5182
4.10.1 GDBRemoteConnection limitations.....	5183
4.11 GenericCounter.....	5184
4.11.1 GenericCounter - parameters.....	5184
4.12 GenericTrace.....	5185
4.12.1 GenericTrace - parameters.....	5186
4.12.2 GenericTrace plug-in usage example.....	5187
4.12.3 Mapping between SYSREG_UPDATE trace sources and SPSR registers.....	5189
4.13 ListTraceSources.....	5190
4.13.1 ListTraceSources - parameters.....	5190
4.14 PipelineModel.....	5191
4.14.1 CortexA53PipelineModel - parameters.....	5192
4.14.2 InOrderPipelineModel - parameters.....	5192
4.14.3 PipelineModel example.....	5193
4.14.4 Naming the plug-in instance.....	5194
4.14.5 Example command lines.....	5194
4.14.6 PipelineModel output.....	5196
4.15 RunTimeParameterTest.....	5197
4.16 Sidechannel.....	5197
4.17 TarmacText.....	5197
4.17.1 TarmacText - parameters.....	5198
4.18 TarmacTrace.....	5199
4.18.1 TarmacTrace - parameters.....	5199
4.18.2 TarmacTrace file format.....	5202
4.18.3 Tarmac Trace output example.....	5203
4.18.4 Instruction trace.....	5205
4.18.5 Program flow trace.....	5207
4.18.6 Register trace.....	5208
4.18.7 Cache maintenance trace.....	5209
4.18.8 Cache content trace.....	5210
4.18.9 Translation table walk trace.....	5211
4.18.10 Granule protection table walk trace.....	5212
4.18.11 TLB trace.....	5213
4.18.12 Event trace.....	5215

4.18.13 Processor memory access trace.....	5216
4.18.14 Processor memory update trace.....	5217
4.18.15 Memory bus trace.....	5219
4.19 ToggleMTIPlugin.....	5221
4.19.1 ToggleMTIPlugin - parameters.....	5221
4.19.2 How to use ToggleMTIPlugin.....	5222
<b>5. Fast Models examples.....</b>	<b>5224</b>
5.1 LISA examples.....	5224
5.2 Build and run an FVP example.....	5225
5.3 LISAPlus examples.....	5229
5.4 MTI examples.....	5229
5.5 SystemCEExport examples.....	5230
5.6 Build and run an EVS platform example.....	5231
5.7 Build and run an SVP example.....	5232
<b>Proprietary notice.....</b>	<b>5234</b>
<b>Product and document information.....</b>	<b>5236</b>
Product status.....	5236
Revision history.....	5236
Conventions.....	5238
<b>Useful resources.....</b>	<b>5241</b>



# 1. About the models

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

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

## 1.1 Model capabilities

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

Fast Models can:

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

Fast Models cannot:

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

### Related information

- [Caches in Fast Models](#)

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

Fast Models does not support running the entire STL.

The reasons are:

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

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

### Related information

- [Arm Software Test Libraries](#)

## 1.3 Quality level definitions

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

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

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

## 1.4 Fast Models accuracy

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

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

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

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

### 1.4.1 Timing accuracy of Fast Models

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

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

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

The consequences of this are:

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



This might mask some race-condition bugs in software.

---

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

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

### 1.4.2 Bus traffic in Fast Models

PVBus can simulate the behavior of individual bus transactions passing through a hierarchy of bus fabric, but it uses the following techniques to optimize this process:

- PVBus generally decodes the path between a bus master and bus slave the first time a transaction is issued. All subsequent transactions to the same address are automatically sent to the same slave, without passing through the intervening fabric.

- For accesses to normal memory, the master can cache a pointer to the host storage that holds the data contents of the memory. The master can read and write directly to this memory without generating bus transactions.
- For instruction-fetch, and for operations such as repeated DMA from framebuffer memory, PVBUS provides an optimization called snooping, which informs the master if anyone else could have modified the contents of memory. If no changes have occurred, the master can avoid the need to re-read memory contents.

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



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

---

The bus traffic generated by a processor is not representative of real traffic:

### Timing differences

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

### Bus contention

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

### Size of access

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

### Instruction fetch

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

### Behavioral differences

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

### Other differences

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

## Related information

[Instruction prefetch in Fast Models](#) on page 25

### 1.4.3 Instruction prefetch in Fast Models

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

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

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

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

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

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

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

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

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

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

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

### 1.4.5 Caches in Fast Models

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

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

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

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

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

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

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

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

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

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

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

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

### 1.4.6 Global exclusive monitor in Fast Models

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

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

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

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

## 1.5 Processor implementation

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

### 1.5.1 Mapping PMU events to MTI trace sources

In Fast Models, many PMU events are modeled and are exposed through the standard PMU interface, although in many cases they are implemented using MTI trace sources.



Note

- Counters that are modeled can generate PMU overflow interrupts, which work the same as in hardware.
- Counters that are not modeled have the value zero and do not increment.

The following tables show which PMUv1 and PMUv2 events are modeled, and which MTI trace sources are used to model them. The associated event numbers are as listed in the Arm® Architecture Reference Manual Arm®v7-A and Arm®v7-R edition, section C12.8.2 Common event numbers.

**Table 1-2: PMUv1 event mappings**

PMUv1 event	MTI trace sources	Notes
Software increment (0x00)	-	Modeled, but not using MTI.
Level 1 instruction cache refill (0x01)	ALLOC_LINEFILL	-
Level 1 instruction TLB refill (0x02)	MMU_TLB_FILL	-
Level 1 data cache refill (0x03)	ALLOC_LINEFILL	-
Level 1 data cache access (0x04)	CACHE_READ_HIT, CACHE_READ_MISS, CACHE_WRITE_HIT, CACHE_WRITE_MISS	-
Level 1 data TLB refill (0x05)	MMU_TLB_FILL	-
Load (0x06)	CORE_LOADS, NV2_CORE_LOADS, MEMTAG_LOAD_INST, SVE_LD_RETIRED	-
Store (0x07)	CORE_STORES, NV2_CORE_STORES, MEMTAG_STORE_INST, SVE_ST_RETIRED	-
Instruction architecturally executed (0x08)	-	Modeled, but not using MTI.

PMUv1 event	MTI trace sources	Notes
Exception taken (0x09)	EXCEPTION	-
Exception return (0x0A)	EXCEPTION_RETURN_PREBRANCH	-
Write to CONTEXTIDR (0x0B)	CONTEXTIDR	-
Software change of the PC (0x0C)	BRA_DIR, BRA_INDIR.	-
Immediate branch (0x0D)	BRA_DIR, BRA_DIR_FAIL.	-
Procedure return (0x0E)	BRA_RET	-
Unaligned load or store (0x0F)	UNALIGNED_LDST_RETIRED, SVE_UNALIGNED_LDST_RETIRED	-
Mispredicted or not predicted branch speculatively executed (0x10)	BRANCH_MISPREDICT	Only if the BranchPrediction plug-in is loaded.
Cycle count (0x11)	-	Modeled, but not using MTI.  <b>Note:</b> This value is an approximation.
Predictable branch speculatively executed (0x12)	-	Not modeled.

**Table 1-3: PMUv2 event mappings**

PMUv2 event	MTI source	Notes
Data memory access (0x13)	-	Not modeled.
Level 1 instruction cache access (0x14)	CACHE_READ_HIT, CACHE_READ_MISS	-
Level 1 data cache write-back (0x15)	-	Not modeled.
Level 2 data cache access (0x16)	CACHE_READ_HIT, CACHE_READ_MISS, CACHE_WRITE_HIT, CACHE_WRITE_MISS	-
Level 2 data cache refill (0x17)	ALLOC_LINEFILL	-
Level 2 data cache write-back (0x18)	-	Not modeled.
Bus access (0x19)	CORE_LOADS, CORE_STORES, MEMTAG_LOADS, MEMTAG_STORES	-
Memory error (0x1A)	-	Not modeled.
Instruction speculatively executed (0x1B)	-	Not modeled.
Write to TTBR (0x1C)	TTBR_WRITE	-
Bus cycle (0x1D)	INST	This value is an approximation.

## Related information

[ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition](#)



## 1.5.2 Caches in PV models

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

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

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

The registers affected on all code translation processor models are:

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

### 1.5.2.1 Functional caches in Fast Models

Fast Models implement two types of cache model: register model and functional model.

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

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

Fast Models provide:

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

**Note**

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

For a core with no L2 cache, the configuration parameters are:

**icache-state-modelled**

Set whether the I-cache has stateful implementation.

**dcache-state-modelled**

Set whether the D-cache has stateful implementation.

For Arm®v7-A cores with an L2 cache, the configuration parameters are:

**l1\_icache-state-modelled**

Set whether L1 I-cache has stateful implementation.

**l1\_dcache-state-modelled**

Set whether L1 D-cache has stateful implementation.

**l2\_cache-state-modelled**

Enable unified Level 2 cache state model.

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

**icache-state\_modelled**

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

**dcache-state\_modelled**

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

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

Enabling functional cache modeling is likely to reduce performance.

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

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

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

### 1.5.3 GICv3 in PV models

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

The GICv3 architecture defines two parts:

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

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

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

Models have the following limitations:

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

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

### 1.5.4 GICv4 in PV models

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

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

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

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

## 1.5.5 CP14 Debug coprocessor

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

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

## 1.5.6 TLBs in PV models

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



Note

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

These TLB registers do not have working implementations:

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

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



Note

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

## 1.5.7 Memory access in PV models

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

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

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

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

### 1.5.7.1 I-side access in PV models

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

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

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

### 1.5.7.2 D-side access in PV models

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

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

Slaves declared as type `pv::DEVICE` are normally used for peripheral access.

## 1.5.8 Timing in PV models

Programmers View (PV) models are loosely timed.

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

## 1.6 syncLevel definitions

Definitions for the possible syncLevel values.

### syncLevel 0

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

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

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

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

### syncLevel 1

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

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

Quantum end detection is as for syncLevel 0.

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

### syncLevel 2

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

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

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

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

### syncLevel 3

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

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

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

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

## Related information

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

## 1.7 Controlling and observing the syncLevel

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

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

This is the exposed interface to control and observe the value of syncLevel. All these registers are in the CADI/SCADI register group `simulation` for each CT processor that contains non-architectural, simulator-specific registers. All are 32-bit integer registers. Users of syncLevel write to these registers to register and unregister for the syncLevel they require:

### **syncLevelSyncStateRegister**

Users write to this register for `SYNC_STATE`. Write-only.

### **syncLevelSyncStateUnregister**

Users write this to unregister for `SYNC_STATE`. Write-only.

### **syncLevelPostInsnIORegister**

Users write to this register for `POST_INSN_LDST`. Write-only.

### **syncLevelPostInsnIOUnregister**

Users write this to unregister for `POST_INSN_LDST`. Write-only.

### **syncLevelPostInsnAllRegister**

Users write this to register for `POST_INSN_ALL`. Write-only.

### **syncLevelPostInsnAllUnregister**

Users write this to unregister for `POST_INSN_ALL`. Write-only.

These registers are only for debugging and visibility in the debugger, and syncLevel users do not usually access them at all:

**syncLevel**

Current syncLevel. Read-only.

**syncLevelSyncStateCount**

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

**syncLevelPostInsnIOCount**

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

**syncLevelPostInsnAllCount**

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

**minSyncLevel**

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

The `syncLevelxxxRegister` and `syncLevelxxxUnregister` registers are for syncLevel users to register and unregister themselves, by writing the value 0 to them. Changes to the syncLevel become effective at the next stop event checkpoint. In addition, syncLevel users can write to these registers any time before the simulation is running, for example from the `init()` simulation phase. The change takes effect immediately when the simulation is run.

The other registers are only present to make the debugging of these mechanisms and their users easier. The `syncLevel` register enables you to see what kind of performance you can expect from the model. You must treat access to these other registers as read-only. You can write to them, however, to permit debugging the syncLevel mechanisms.

These registers are not memory (or CPnn-) mapped anywhere, and are not accessible to target programs.

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

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

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

You can change this parameter at runtime, and changes become effective at the next stop event checkpoint. Reading this parameter value returns the `min_sync_level`, not the current syncLevel. This parameter is only an additional way of controlling the syncLevel and controls the same mechanisms as the register interface.



## 1.8 User mode networking

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

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

### SMSC\_91C111

This is paired with an external HostBridge component. The user mode networking specification is set on the external HostBridge.

### VirtioNetMMIO

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



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

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

### SMSC\_91C111:

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

### VirtioNetMMIO:

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

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

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

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

**SMSC\_91C111:**

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

**VirtioNetMMIO:**

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

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

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

The default is to accept connections on any interface.

**Related information**

- [HostBridge](#)

## 1.9 TAP/TUN networking

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

### 1.9.1 TAP/TUN networking limitations

TAP/TUN networking on Fast Models has these limitations.

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

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

This section describes how to set up a network connection.

**Before you begin**

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

## About this task

---



Note

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

### Procedure

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

#### 32-bit operating system

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

#### 64-bit operating system

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

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



Note

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

---

## Next steps

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

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

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

## 1.9.3 Setting up a network connection for Ubuntu Linux

This section describes how to set up a network connection.

### About this task

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



Note

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

---

### Procedure

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

Use one of the following ways:

- For an interface using DHCP, add:

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

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

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

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

2. Install the bridge-utils package:

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

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

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

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

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

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

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

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

This step creates a TAP device for each user.

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

```
bridge_ports eth0 ARMfmuser
```

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

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

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

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

**Note**

armbr0 must be explicitly started.

This step disconnects and reconnects all network interfaces.

## 1.9.4 Configuring the networking environment for Linux

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

### Before you begin

Use System Canvas or a related Fast Models tool to load a project or model, and then select a component.

### About this task

**Note**

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

### Procedure

Set the parameters on the `HostBridge` and `SMSC_91C111` components, or on the `VirtioNetMMIO` component and its `HostBridge` subcomponent. For example:

#### SMSC\_91C111:

```
hostbridge.interfaceName=ARM<username>
smcsc_91c111.enabled=1
```

#### VirtioNetMMIO:

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

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

### Related information

[Fast Models Tools User Guide](#)

## 1.9.5 Solutions to networking issues on Linux

This section describes how to solve networking issues.

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

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

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

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

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

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

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



Note

Refer to the vendor's documentation manual.

---

## 1.9.6 Disabling and re-enabling networking for Linux

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

### About this task



Caution

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

---

### Procedure

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

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

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

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

## 1.9.7 Uninstalling networking for Linux

This section describes the steps to uninstall a network.

### Procedure

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

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

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



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

---

### Next steps

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

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

## 1.10 Using parameters to set port values

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

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

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



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

---

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



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

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

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

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

### 1.11.1 Class `pv::TransactionGenerator`

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

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

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

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

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

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

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

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

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

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

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

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

### 1.11.2 TransactionGenerator efficiency considerations

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

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

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

### 1.11.3 Enum `pv::AccessWidth`

This enum selects the required bus width for a transaction.

Defined values are:

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

### 1.11.4 Class `pv::Transaction`

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

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

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

```

class pv::Transaction
{
public:
    // Accessors
    bus_addr_t getAddress() const;           // Transaction address.
    pv::AccessWidth getAccessWidth() const;  // Request width.
    int getAccessByteWidth() const;          // Request width in bytes.
    int getAccessBitWidth() const;           // Request width in bits.

    bool isPrivileged() const;               // Privileged process mode?
    bool isInstruction() const;              // Instruction request vs data?
    bool isNonSecure() const;               // Normal-world vs secure-world?
    bool isLocked() const;                  // Atomic locked access?

```

```

bool isExclusive() const; // Atomic exclusive access?
uint32_t getMasterID() const;
bool hasSideEffect() const;

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

```

### 1.11.5 Class pv::ReadTransaction

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

```

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

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

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

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

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

```

### 1.11.6 Class pv::WriteTransaction

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

```

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

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

```

```

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

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

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

```

## 1.12 Visualisation library

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

The types of I/O handled include:

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

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

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

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

### Related information

[LCD protocol](#) on page 84

[LISA visualisation models](#) on page 49

[Visualisation library C++ classes](#) on page 49

## 1.12.1 LISA visualisation models

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

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

### Related information

[Versatile Express model](#)

## 1.12.2 Visualisation library C++ classes

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

### 1.12.2.1 C++ classes inclusion

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

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

### 1.12.2.2 Class Visualisation

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

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

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

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

The `visualisation` class has the following methods:

#### **~Visualisation()**

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

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

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

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

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

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

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

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

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

**void releaseImage(VisBitmap \*)**

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

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

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

**VisRenderRegion \*createRenderRegion()**

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

**VisPushButtonRegion \*createPushButtonRegion()**

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

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

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

The method returns false if no events have occurred.

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

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

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

**void unlockMouse()**

Unlocks and redisplay the mouse pointer.

**bool hasQuit()**

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

### 1.12.2.3 Class VisRegion

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

#### **~VisRegion()**

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

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

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

#### **void \*getId()**

Returns the client-defined identifier.

#### **void setVisible(bool vis)**

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

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

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

### 1.12.2.4 Class VisPushButtonRegion : public VisRegion

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

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

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

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

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

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

- `void setKeyCode(intcode)`

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

### 1.12.2.5 Class VisRenderRegion : public VisRegion

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

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

**VisRasterLayout const \*lock()**

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

**void unlock()**

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

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

Causes the specified rectangle to be drawn to the GUI.

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

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

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

Draws a bitmap onto an unlocked VisRenderRegion.

### 1.12.2.6 Struct VisRasterLayout

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

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

The structure contains these fields:

**uint8\_t\* buffer**

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

**uint32\_t pitch**

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

**uint32\_t width**

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

**uint32\_t height**

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

**VisPixelFormat format**

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



**bool changed**

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

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

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

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

### 1.12.2.7 Struct VisPixelFormat

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

The members are:

**uint32\_trbits, gbits, bbits**

The number of bits per color channel.

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

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

**uint32\_t pbytes**

The size of a single pixel, in bytes.

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

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

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

## 2. Protocols

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

### 2.1 AMBAPV protocol

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

#### About AMBAPV protocol

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

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

- Protection units.
- Exclusive access and locked access mechanisms.
- System-level caches.
- Atomic accesses, including exclusive accesses, locked accesses, and atomic transactions.

It always returns the original data at the target address.

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

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

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

AMBAPV provides the following behaviors:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**debug\_read()**

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

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

**debug\_write()**

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

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

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

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

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

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

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

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

**read()**

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

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

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

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

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

### **write()**

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

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

## **2.2 AMBAPVACE protocol**

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

### **About AMBAPVACE protocol**

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

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

- Secure and privileged accesses.
- Atomic accesses.
- System-level caching and buffering control.
- Cache coherency transactions (ACE-Lite).
- Bi-directional cache coherency transactions (ACE).
- Distributed virtual memory transactions (DVM).

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

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

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

### Special considerations for ACE and cache coherent interconnects

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

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

#### debug reads

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

#### debug writes

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

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

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

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

AMBA PVACE provides the following behaviors:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 2.3 AMBAPVSignal protocol

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

### About AMBAPVSignal protocol

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

`AMBAPVSignal` provides the following behaviors:

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

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

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

## 2.4 AMBAPVSignalState protocol

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

### About AMBAPVSignalState protocol

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

`AMBAPVSignalState` provides the following behaviors:

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

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

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

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

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

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



## 2.5 AMBAPVValue protocol

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

### About AMBAPVValue protocol

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

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

AMBAPVValue provides the following behaviors:

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

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

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

## 2.6 AMBAPVValue64 protocol

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

### About AMBAPVValue64 protocol

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

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

AMBAPVValue64 provides the following behaviors:

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

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

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

## 2.7 AMBAPVValueState protocol

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

### About AMBAPVValueState protocol

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

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

AMBAPVValueState provides the following behaviors:

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

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

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

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

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

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

## 2.8 AMBAPVValueState64 protocol

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

### About AMBAPVValueState64 protocol

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

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

AMBAPVValueState64 provides the following behaviors:

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

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

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

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

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

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

## 2.9 AsyncSignalCallback protocol

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

### About AsyncSignalCallback protocol

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

AsyncSignalCallback provides the following behaviors:

#### signal()

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

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

This is called asynchronously after a non-simulation thread has called AsyncSignal.async\_control.signal().

## 2.10 AsyncSignalControl protocol

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

### About AsyncSignalControl protocol

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

AsyncSignalControl provides the following behaviors:

#### signal()

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

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

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

## 2.11 AudioControl protocol

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

### About AudioControl protocol

This protocol has get and release audio buffer behaviors.

AudioControl provides the following behaviors:

### **getPVAudioBuffer()**

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

Get an underlying host buffer for audio output.

### **releasePVAudioBuffer()**

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

Release an underlying host buffer.

## 2.12 CADIDisassemblerProtocol protocol

Defined in \$PVLIB\_HOME/LISA/CADIDisassemblerProtocol.lisa.

### About CADIDisassemblerProtocol protocol

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

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

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

CADIDisassemblerProtocol provides the following behaviors:

### **GetAddressForSourceReference()**

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

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

### **GetCurrentMode()**

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

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

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

### **GetDisassembly()**

```
slave behavior GetDisassembly(eslapi::CADIDisassemblerCB *callback_,
```

```

                                const eslapi::CADIAddr_t &address,
                                eslapi::CADIAddr_t &nextAddr,
                                const uint32_t mode,
                                uint32_t desiredCount) :
    eslapi::CADIDisassemblerStatus;

```

The main disassembler function for standard type disassembly.

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

### **GetInstructionType()**

```

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

```

Query if an instruction is a call instruction.

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

### **GetModeCount()**

```

slave behavior GetModeCount() : uint32_t;

```

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

### **GetModeNames()**

```

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

```

Query the names of all supported modes.

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

### **GetSourceReferenceForAddress()**

```

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

```

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

### **GetType()**

```

slave behavior GetType() : eslapi::CADIDisassemblerType;

```

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

## 2.13 CADIProtocol protocol

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

### About CADIProtocol protocol

This protocol supports debugging.

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

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

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

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

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

`CADIProtocol` provides the following behaviors:

#### **CADIBptClear()**

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

Clear the breakpoint specified by `CADIBptNumber_t`.

#### **CADIBptConfigure()**

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

Re-configure an existing breakpoint.

**CADIBptGetList()**

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

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

**CADIBptRead()**

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

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

**CADIBptSet()**

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

Create a new breakpoint. The breakpoint number is returned.

**CADIExecSingleStep()**

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

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

**CADIGetCycleCount()**

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

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



Note

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

**CADIGetDisassembler()**

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

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

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

Skeleton code for implementing disassembly:

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

### **CADIGetInstructionCount()**

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

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

### **CADIModifyTargetFeatures()**

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

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

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

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



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

```
targetFeatures->handledBreakpoints = CADI_TARGET_FEATURE_BPT_PROGRAM |
    CADI_TARGET_FEATURE_BPT_REGISTER;

targetFeatures->nrBreakpointsAvailable = 0x7fffffff;
```

### **callbackModeChange ()**

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

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

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

## 2.14 CCI500\_AddressDecoderProtocol protocol

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

### About CCI500\_AddressDecoderProtocol

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

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

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

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

### Limitations

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

CCI500\_AddressDecoderProtocol provides the following behaviors:

### configuration()

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

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

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

### decode\_by\_4KiB\_addr()

```
slave behavior decode_by_4KiB_addr( unsigned upstream_port_index_,
                                   bool      is_read_or_cmo_,
                                   uint64_t address_,
                                   bool      ns_ ) : unsigned;
```



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

### get\_squash\_record()

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

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

//! addr[out\_lsb\_bitpos\_ + out\_number\_of\_bits\_to\_squash\_ : out\_lsb\_bitpos\_] //! "

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

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

### reset()

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

This is called when CCI-500 is reset.

## 2.15 CCIInterconnectControl protocol

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

### About CCIInterconnectControl

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

CCIInterconnectControl provides the following behaviors:

#### getControlIf()

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

Get CCIInterconnect control\_if pointer.

## 2.16 ClockRateControl protocol

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

### About ClockRateControl protocol

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

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

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

ClockRateControl provides the following behaviors:

#### set()

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

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

#### set64()

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

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

## 2.17 ClockSignal protocol

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

### About ClockSignal protocol

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

ClockSignal output ports are provided on the following library components:

#### MasterClock

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

#### ClockDivider

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

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



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

Here is an example system using ClockSignals:

```

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

```

clocksignal provides the following behaviors:

**currentTicks()**

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

Private internal method used between Scheduler components.

**getClock()**

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

Private internal method used between Scheduler components.

**rateInHz()**

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

Private internal method used between Scheduler components.

**setClock()**

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

Private internal method used between Scheduler components.

## 2.18 CompoundPortLisa protocol

Defined in \$PVLIB\_HOME/LISA/CompoundPort.lisa.

CompoundPortLisa provides the following behaviors:

**connectFromExternalSlavePort()**

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

**connectToExternalMasterPort()**

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

**disconnectFromExternalSlavePort()**

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

**disconnectToExternalMasterPort()**

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

## 2.19 CoprocBusProtocol protocol

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

### About CoprocBusProtocol protocol

This protocol connects a coprocessor implementation with a CPU component, for instance ARM CortexM33CT.

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

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

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

### Registering a coprocessor

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

...
    };

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

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

## Coprocessor callback functions

A coprocessor can implement callback functions with these signatures.

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

### **CDP()**

Perform a coprocessor data processing operation.

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

Parameters:

#### **context**

Context that was registered with the coprocessor interface.

#### **inst**

The coprocessor instruction being executed.

### **MCR()**

Perform a move to coprocessor register operation.

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

Parameters:

#### **context**

Context that was registered with the coprocessor interface.

#### **inst**

The coprocessor instruction being executed.

#### **data**

Register contents.

### **MRC()**

Perform a move from coprocessor register operation.

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

Parameters:

#### **context**

Context that was registered with the coprocessor interface.

#### **inst**

The coprocessor instruction being executed.

#### **data**

Pointer to word to fill with coprocessor register contents.

### **LDC()**

Perform a load coprocessor register from memory operation.

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

Parameters:

**context**

Context that was registered with the coprocessor interface.

**inst**

The coprocessor instruction being executed.

**data**

Contents of current memory location to load into register.

**state**

Current state in a sequence of transactions.

**STC ()**

Perform a store coprocessor register to memory operation.

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

Parameters:

**context**

Context that was registered with the coprocessor interface.

**inst**

The coprocessor instruction being executed.

**data**

Pointer to word to fill with coprocessor register contents to be transferred to memory.

**state**

Current state in a sequence of transactions.

**MCRR ()**

Perform a move to two coprocessor registers operation.

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

Parameters:

**context**

Context that was registered with the coprocessor interface.

**inst**

The coprocessor instruction being executed.

**data1**

First data word to load to a coprocessor register.

**data2**

Second data word to load to a coprocessor register.

**MRRC ()**

Perform a move from two coprocessor registers operation.

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

Parameters:



**context**

Context that was registered with the coprocessor interface.

**inst**

The coprocessor instruction being executed.

**data1**

Pointer to first word to fill with coprocessor register contents.

**data2**

Pointer to second word to fill with coprocessor register contents.

**CoprocState values**

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

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

CoprocBusProtocol provides the following behaviors:

**accessIsNonSecure()**

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

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

**accessIsPriv()**

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

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

**addCoprocessor()**

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

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

**removeCoprocessor()**

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

Unregisters the coprocessor from the CPU.

## 2.20 CounterInterface protocol

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

### About CounterInterface protocol

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

CounterInterface provides the following behaviors:

#### **eventUpdate()**

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

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

#### **getCounterValue()**

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

Get the absolute value of the physical timer.

#### **requestEventUpdate()**

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

Request an eventUpdate at a particular time.

#### **requestSignalUpdate()**

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

Request a signalUpdate at a particular time.

#### **setEnabled()**

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

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

#### **signalUpdate()**

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

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

## 2.21 DVMMMessage protocol

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

`DVMMMessage` provides the following behaviors:

### **send()**

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

Send DVM message.

## 2.22 EventBus protocol

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

`EventBus` provides the following behaviors:

### **publishEventSource()**

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

## 2.23 Feature protocol

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

`Feature` provides the following behaviors:

### **setFeature()**

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

Set feature `featureName` to `valueStr`.

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

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

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

## 2.24 FlashLoaderPort protocol

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

### About FlashLoaderPort

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

FlashLoaderPort provides the following behaviors:

#### loadFlashFile()

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

Initiate loading of the flash contents.

#### saveFlashFile()

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

Save the flash contents to a file.

## 2.25 FrameTracingProtocol protocol

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

### About FrameTracingProtocol protocol

Port type used to connect to a FrameTracingComponent.

FrameTracingProtocol provides the following behaviors:

#### beginFrame()

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

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



Note

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

**endFrame()**

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

Notify the downstream FrameTracingComponent that the buffer provided by beginFrame() is now blitted with a frame and can be processed.

## 2.26 GICv3Comms protocol

Defined in \$PVLIB\_HOME/LISA/GICv3Comms.lisa.

**About GICv3Comms protocol**

Link for internal communications between GICv3 components.

The master is towards the top level, the slave is towards the CPU interface.

GICv3Comms provides the following behaviors:

**sendTowardsCPU()**

```
slave behavior sendTowardsCPU(uint8_t len, const uint8_t* data) : void;
```

Sends byte stream towards the core.

**sendTowardsTopLevel()**

```
master behavior sendTowardsTopLevel(uint8_t len, const uint8_t* data) : void;
```

Sends byte stream from the core.

**setAXIMasterID()**

```
master behavior setAXIMasterID(uint32_t master_id) : void;
```

Sets the MasterID associated with the stream from the core.

## 2.27 GUIPollCallback protocol

Defined in \$PVLIB\_HOME/LISA/GUIPollCallbackPort.lisa.

**About GUIPollCallback protocol**

Callback signal generated by a GUIPoll component. It allows a Visualisation component to continue to poll the GUI's event queue while the simulation is paused.

See the [GUIPoll] component for advice about using this protocol.

GUIPollCallback provides the following behaviors:

### **gui\_callback()**

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

Client callback invocation, called at a period configured by the GUIPoll component.



Note

This callback should only be used for updating a visualisation GUI. It should never be used for simulation events. See [GUIPoll] for more information.

## 2.28 ICS307Configuration protocol

Defined in \$PVLIB\_HOME/LISA/ICS307ConfigurationPort.lisa.

### **About ICS307Configuration protocol**

This protocol sets the divider ratio of an ICS307 component at runtime. The output clock rate alters accordingly and any dependent components react to the clock rate change according to their defined behavior.

ICS307Configuration provides the following behaviors:

### **setConfiguration()**

```
peer behavior setConfiguration(uint32_t vdw, uint32_t rdw, uint32_t od) : void;
```

Set the parameters for deriving the clock divider ratio.

**vdw**

Range: 0-255.

**rdw**

Range: 0-255.

**od**

Range: 0-7.

## 2.29 InstructionCount protocol

Defined in \$PVLIB\_HOME/LISA/InstructionCountProtocol.lisa.

InstructionCount provides the following behaviors:

**getRunState()**

```
master behavior getRunState() : uint32_t;
```

Obtain the power/run status of the processor.

Value	State label	Description
0	<b>UNKNOWN</b>	Run status unknown, that is, simulation has not started
1	RUNNING	Processor running, is not idle and is executing instructions
2	HALTED	External halt signal asserted
3	STANDBY_WFE	Last instruction executed was WFE and standby mode has been entered
4	STANDBY_WFI	Last instruction executed was WFI and standby mode has been entered
5	IN_RESET	External reset signal asserted
6	DORMANT	Partial processor power down
7	SHUTDOWN	Complete processor power down

**getValue()**

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

Obtain the number of instructions executed by the processor.

## 2.30 KeyboardStatus protocol

Defined in \$PVLIB\_HOME/LISA/KeyboardStatusProtocol.lisa.

**About KeyboardStatus protocol**

This protocol passes keyboard events to a component such as the PS2Keyboard component.

Events are only sent when the visualization window is in focus. Keyboard combinations that are filtered by the host OS such as Ctrl+Alt+Del are not detected by the visualization.

See \$PVLIB\_HOME/include/components/KeyCode.h for a list of ATKeyCode values.

KeyboardStatus provides the following behaviors:

**keyDown()**

```
slave behavior keyDown(ATKeyCode code) : void;
```

Sent when a key on the host keyboard is pressed.

**keyUp()**

```
slave behavior keyUp(ATKeyCode code) : void;
```

Sent when a key on the host keyboard is released.

## 2.31 LCD protocol

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

### About LCD protocol

This Visualisation Library signaling protocol provides the interface between an LCD controller peripheral, for example the PL110, and a visualization component. This permits the LCD controller to render the framebuffer contents into a region of the visualization GUI.

LISA visualization components can provide any number of LCD ports. The implementations of these behaviors can delegate the calls to appropriate methods on the `VisRenderRegion` class.

`LCD` provides the following behaviors:

#### **lock()**

```
slave behavior lock() : const VisRasterLayout*;
```

Lock the raster region, ready for rendering onto.

#### **setPreferredLayout()**

```
slave behavior setPreferredLayout(unsigned int width, unsigned int height, unsigned  
int depth) : void;
```

Set the preferred pixel size and bitdepth of the LCD panel.

#### **unlock()**

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

Unlock the raster region, ready to update.

#### **update()**

```
slave behavior update(int x, int y, unsigned int w, unsigned int h) : void;
```

Update part of the render region onto the screen.



## 2.32 LCDLayoutInfo protocol

Defined in `$PVLIB_HOME/examples/LISA/Common/LISA/LCDLayoutInfoProtocol.lisa`.

### About LCDLayoutInfo protocol

This protocol has the behavior `setLayoutInfo`.

`LCDLayoutInfo` provides the following behaviors:

#### **setLayoutInfo()**

```
slave behavior setLayoutInfo(int x, int y, uint32_t w, uint32_t h) : void;
```

Sets the width and height of the touchscreen.

## 2.33 MMC\_Protocol protocol

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

### About MMC\_Protocol

This protocol describes an abstract, untimed interface between an MMC controller and an MMC or SD card.

The protocol contains methods that must be implemented by the master (controller) or by the slave (card). This protocol is used by the reference PL180 MCI and MMC models. For further information on the protocol implementation, see the source file `$PVLIB_HOME/LISA/MMC_Protocol.lisa`.

Use of this protocol assumes knowledge of the MultiMediaCard specification, available from the [MultiMediaCard Association](#).

`MMC_Protocol` provides the following behaviors:

#### **Rx()**

```
master behavior Rx(const uint8_t *block, uint32_t len) : bool;
```

Read behaviours, from the card to the controller.

After the controller has issued a block or multiple block read command, the card calls the controller's `Rx()` method, with the first block. When the controller has consumed the block, that is, when it is able to accept another block, it should inform the slave with an `Rx_rdy()` call.

The slave might not provide a block immediately. It might wait until the controller is ready and the simulated transfer rate limits have been satisfied. This is important to avoid swamping the simulation with a large transfer at the expense of all other simulation activity.

The master might signal that it was not able to accept the given block, by returning false from `Rx()`. This is effectively a protocol error, and the card may retransmit the block later, or fail.

### **`Rx_rdy()`**

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

### **`Tx()`**

```
master behavior Tx(uint8_t *block, uint32_t len) : bool;
```

Write behaviours, from the controller to the card.

To minimize the number of times written data are copied, the following protocol is somewhat counter-intuitive. The basic premise is that an MMC controller usually contains a small data FIFO, which is filled either by the simulated CPU, or more frequently by DMA. The DMA typically occurs word by word. An efficient approach is therefore to construct the controller such that it can write directly into a buffer of stored card data.

When the write command is issued, the card calls the master with a pointer to the block that needs to be written. The master can then fill the block, calling `Tx_done()` when the block has been transferred. The card is again responsible for throttling to a simulated transfer rate, and will respond at some time in the future by providing another block to be written by calling the controller's `Tx()` function.

This approach has some drawbacks:

- Some timing and controller behavioral accuracy is sacrificed
- The controller might need to buffer data before a block is provided, if it cannot prevent data coming into its FIFO.

### **`Tx_done()`**

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

### **`cmd()`**

```
slave behavior cmd(mmc_cmd_t cmd, uint32_t arg, void *resp) : mmc_resp_t;
```

The controller can send the slave a command, with an optional 32-bit argument.

The master must send in a void pointer to 128 bits of data.

The slave responds with a response type, and fills in up to 128 bits with data.

The master can check that the response type matches expectations, but this should not be necessary.

CRC is not implemented and start/stop bits are unnecessary at this level.



This behavior is not re-entrant by current design.

### **cmd\_name()**

```
slave behavior cmd_name(mmc_cmd_t cmd) : const char*;
```

The slave implements this behavior to return a string for a given command. This is not strictly part of the MMC protocol.

## 2.34 MMU\_400\_BASE\_IDENTIFY protocol

Defined in \$PVLIB\_HOME/LISA/SMMU\_400\_BASE.lisa.

MMU\_400\_BASE\_IDENTIFY provides the following behaviors:

### **identify()**

```
slave behavior identify(
    const pv::TransactionAttributes* attributes_,
    bool is_read_,
    unsigned* stream_id_,
    unsigned* ssd_or_ssd_index_
) : void;
```

The way that the MMU-400 is configured to generate the `streamID` and `ssd_index` is complicated and must be done by implementing this function. This knowledge is specific to the SoC and to the devices generating the transactions and so it is not easily parameterisable.

## 2.35 MMU\_400\_Internals protocol

Defined in \$PVLIB\_HOME/LISA/SMMU\_400\_BASE.lisa.

### **About MMU\_400\_Internals protocol**

This protocol is for probing the internals of the MMU\_400. It has no correspondence in hardware. It is only intended for testing and informational purposes.

MMU\_400\_Internals provides the following behaviors:

### **getMMU\_400()**

```
slave behavior getMMU_400() : MMU_400::mmu_400_if*;
```

## 2.36 MMU\_500\_BASE\_IDENTIFY protocol

Defined in `$PVLIB_HOME/LISA/SMMU_500_BASE.lisa`.

`MMU_500_BASE_IDENTIFY` provides the following behaviors:

### **identify()**

```
slave behavior identify(
    const pv::TransactionAttributes* attributes_,
    bool is_read_,
    unsigned tbu_number_,
    unsigned* stream_id_,
    unsigned* ssd_or_ssd_index_
) : void;
```

The way that the MMU-500 is configured to generate the StreamID and SSD\_Index or SSD is complicated and must be done by implementing this function. This knowledge is specific to the SoC and to the devices generating the transactions and so it is not easily parameterisable.

Note that the LACr0/r1 RTL encodes the TBU number into the bits [14:10] of the StreamId and the SSD\_Index, if being used. The bottom 10 bits are determined from the incoming transaction. In the LACr0/r1 RTL, each TBU can have fewer than 10 bits of `streamId/ssd_Index`, in which case they are zero-extended before being placed into bits[9:0].

The width of the TBU ID busses is invisible to the programmer and does not have an effect in the model except that this `identify()` function must obey the SoC's configuration.

For LACr0/r1, the caller automatically puts the TBU number in the ids itself. The callee is also allowed to do this, but the result is checked by an `assert()`, otherwise just leave these bits as zero.

For EAC, no such check is made and the platform must supply all 15 bits.

If you are supplying an SSD directly, and set the parameter `use_ssd_determination_table` to false so that `SMMU_IDR1.SSDTP == 0`, then the constants generated by `components/SMMU.h:ssd_secure()` and `ssd_non_secure()` should be used to return the SSD in `*ssd_or_ssd_index_`.

## 2.37 MMU\_500\_Internals protocol

Defined in `$PVLIB_HOME/LISA/SMMU_500_BASE.lisa`.

### **About MMU\_500\_Internals protocol**

This protocol is for probing the internals of the MMU\_500. It has no correspondence in hardware. It is only intended for testing and informational purposes.

`MMU_500_Internals` provides the following behaviors:

**getMMU\_500()**

```
slave behavior getMMU_500() : MMU_500::mmu_500_if*;
```

## 2.38 MouseStatus protocol

Defined in \$PVLIB\_HOME/LISA/MouseStatusProtocol.lisa.

**About MouseStatus protocol**

This protocol passes mouse movement and button events to another component such as the PS2Mouse component.

Events are only sent when the visualization window is in focus.

MouseStatus provides the following behaviors:

**mouseButton()**

```
slave behavior mouseButton(uint8_t button, bool down) : void;
```

This is sent when a button on the host mouse is pressed or released.

`button` indicates which button has been pressed or released and is typically 0, 1, or 2 but can be anything up to 7 depending on the OS and attached mouse.

`down` is true if a button is pressed and false if released.

**mouseMove()**

```
slave behavior mouseMove(int dx, int dy) : void;
```

This is sent when the host mouse is moved. Mouse movement events are always relative.

## 2.39 PASSwitchControl protocol

Defined in \$PVLIB\_HOME/LISA/PASSwitch.lisa.

**About PASSwitchControl protocol**

Allow transactions from the RME world (realm/pas/secure/non\_secure) to be routed separately.

Transactions for the RME PAS worlds are, by default, routed through the manager port `pvbus_m[PAS-value]`, where `PAS-value` is:

**0**

Secure

- 1 Non-secure
- 2 Root
- 3 Realm

The control port allows each PAS world to be one of the following:

- Routed to any of the four manager ports `pvbuse_m[0]...pvbus_m[3]`
- Ignored
- Aborted

`PASSwitchControl` provides the following behaviors:

#### **`routeAccessesForRmeWorlds()`**

```
slave behavior routeAccessesForRmeWorlds(
    pv::PASSwitch_RouteOption route_secure,
    pv::PASSwitch_RouteOption route_non_secure,
    pv::PASSwitch_RouteOption route_root,
    pv::PASSwitch_RouteOption route_realm,
    pv::PASSwitch_RouteOption route_system_agent,
    pv::PASSwitch_RouteOption route_non_secure_protected) : void;
```

The arguments to the control port behavior `routeAccessesForRmeWorlds()` select how the chosen transactions are routed. They can have the following values:

#### **`PORT_IGNORE`**

Transactions are ignored. Reads return 0.

#### **`PORT_ABORT`**

Cause transactions to generate an abort.

#### **`PORT_0`**

Route transactions to `pvbuse_m[0]`.

#### **`PORT_1`**

Route transactions to `pvbuse_m[1]`.

#### **`PORT_2`**

Route transactions to `pvbuse_m[2]`.

#### **`PORT_3`**

Route transactions to `pvbuse_m[3]`.

#### **`PORT_4`**

Route transactions to `pvbuse_m[4]`.

#### **`PORT_5`**

Route transactions to `pvbuse_m[5]`.

Initial routing is configured using these PASSwitch parameters:

- `secure_port_index`
- `non_secure_port_index`
- `root_port_index`
- `realm_port_index`

Both default and explicit parameter values are overridden by runtime calls to `routeAccessesForRmeWorlds()` on the control port.

## 2.40 PCIDevice2ClientProtocol protocol

Defined in `$PVLIB_HOME/LISA/PCIDevice2ClientProtocol.lisa`.

### About PCIDevice2ClientProtocol protocol

This is a private protocol between `PCIDevice` and its wrapped client device.

`PCIDevice2ClientProtocol` provides the following behaviors:

#### **`check_if_msix_is_enabled()`**

```
optional master behavior check_if_msix_is_enabled() : bool;
```

A request from the client device to know whether MSI-X interrupt generation capability is enabled for the endpoint.

#### **`generate_MSI_X()`**

```
master behavior generate_MSI_X(
    unsigned          vector_index_,
) : int;
```

A request from the client device to map an MSI vector address to an address and data. Returns false if no MSI should be generated.

Return values:

- 1**  
Abort
- 0**  
Suppressed
- 1**  
OK

**get\_PRI\_client\_interface()**

```
slave behavior get_PRI_client_interface() : pcie_client_device_pri_if*;
```

If the device has PRI capability, it can adjust how the ATC makes PRI requests by implementing this interface.

**get\_transaction\_monitor\_control\_if()**

```
optional master behavior get_transaction_monitor_control_if(
    pcie::pcie_transaction_monitor_client_if* client_if_
) : pcie::pcie_transaction_monitor_control_if*;
```

A protocol which a client device can use to get the transaction monitor control interface implemented by the endpoint. The client device does this by requesting the interface using `get_transaction_monitor_control_if()`. While requesting, the client device can pass a pointer to its own interface, which then provides an interface from the endpoint to the client.

**identify()**

```
optional slave behavior identify(const pv::RemapRequest& req_, uint32_t*
    substreamid_) : void;
```

If the client can produce substreamids, it must use this behavior to fill `substreamid_`. If no substreamid is present on the request represented by `req_` then it should be assigned to `~0u`.

**log\_error()**

```
optional master behavior log_error(
    pcie_service::ErrorMessage::ErrorCode_t error_code_,
    pcie_service::pcie_aer_error_type_t error_type_) : void;
```

A request to log a client device error with a specific error message.

## 2.41 PCIeATC\_get\_if protocol

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

PCIeATC\_get\_if provides the following behaviors:

**get\_if()**

```
slave behavior get_if() : pcie_atc_if*;
```



## 2.42 PChannel protocol

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

### About PChannel protocol

Communicates power state changes between a power controller and a device.

You can use PChannels to replace `STANDBYWFI` and `STANDBYWFE` signaling.

For example, using `STANDBYWFI` or `STANDBYWFE`:

- Core drives `STANDBYWFI` signal HIGH.
- Power controller performs logic x.

Equivalent behavior using PChannels:

- Core calls `pactive(OFF)`.
- Power controller calls `prequest(OFF)` to change the core to OFF.
- Power controller performs logic x.
- To wake up the core, the power controller calls `prequest(ON)`.

### Examples

- For a LISA+ example that uses PChannel, see `$PVLIB_HOME/examples/LISA/VP_PChannel/`.
- For a SystemC example that uses PChannel, see `$PVLIB_HOME/examples/SystemCExport/EVS_Components/EVS_PChannel/`.

`PChannel` provides the following behaviors:

#### **`pactive()`**

```
master behavior pactive (uint32_t pstate) : void;
```

This master behavior is implemented by a power controller. A device calls this method to give a hint to the power controller that it can change to a particular power state. A power controller can then take appropriate action, typically communicating with the device by calling `device.prequest(new_power_state)`.

The power state is type `uint32_t` because it is the responsibility of the system using PChannels to enumerate the power states that it supports. For example, Armv8-A cores use the following enumeration for power states:

```
enum { OFF = 0,
      OFF_EMU,
      MEM_RET,
      MEM_RET_EMU,
      LOGIC_RET,
      FULL_RET,
      MEM_OFF,
      FUNC_RET,
```

```
ON,
WARM_RST,
DBG_RECOV }
```

### **prequest()**

```
slave behavior prequest (uint32_t pstate) : sg::PChannel::presp_t;
```

This slave behavior is implemented by a device, for instance a core. A power controller typically calls this method and checks for the response from the device, which can either be `ACCEPT` or `DENY`.

The `sg::PChannel::presp_t` enumeration provides two values, `ACCEPT` and `DENY`. It is returned by the `prequest()` method, depending on the state requested and the current state of the core.

## 2.43 PL080\_DMAC\_DmaPortProtocol protocol

Defined in `$PVLIB_HOME/LISA/PL080_DMAC_DmaPortProtocol.lisa`.

### About PL080\_DMAC\_DmaPortProtocol protocol

The `DmaPortProtocol` is used to communicate handshake signals between the `PL080_DMAC` controller and other peripherals in the system.

Depending on the `PL080_DMAC` configuration, the `PL080_RES_CLR` signal might not be used.

`PL080_DMAC_DmaPortProtocol` provides the following behaviors:

### **request()**

```
slave behavior request(uint32_t request) : void;
```

Passes requests from a peripheral to the DMA controller. The request is a bitfield with the low four bits defined. The request is level-sensitive and latched internally by the DMA controller. It is sampled and interpreted in a manner dependent on the target channel and configured flow control. It can have one of the following values:

- 1  
    `PL080_REQ_BURST`. Burst transfer request.
- 2  
    `PL080_REQ_SINGLE`. Single transfer request.
- 4  
    `PL080_REQ_LBURST`. Last burst request.
- 8  
    `PL080_REQ_LSINGLE`. Last single request.

**response()**

```
master behavior response(uint32_t response) : void;
```

Passes responses from the DMA controller to the peripherals. The response is a bitfield with the low two bits defined. It is transient rather than level-sensitive:

**1**

PL080\_RES\_TC. Terminal count response.

**2**

PL080\_RES\_CLR. Clear request response.

## 2.44 PL330CppToLISA protocol

Defined in \$PVLIB\_HOME/LISA/PL330\_DMAC.lisa.

PL330CppToLISA provides the following behaviors:

**cancel\_wakeup\_timer()**

```
peer behavior cancel_wakeup_timer() : void;
```

**message\_printer()**

```
peer behavior message_printer( uint32_t type_, std::string text_ ) : void;
```

**update\_abort()**

```
peer behavior update_abort( bool state_ ) : void;
```

**update\_irq()**

```
peer behavior update_irq( unsigned index, bool state_ ) : void;
```

**wakeup\_in\_ticks()**

```
peer behavior wakeup_in_ticks( unsigned ticks_ ) : void;
```

## 2.45 PL330\_DMAC\_DmaPortProtocol protocol

Defined in \$PVLIB\_HOME/LISA/PL330\_DMAC\_DmaPortProtocol.lisa.

**About PL330\_DMAC\_DmaPortProtocol protocol**

The DmaPortProtocol is used to communicate handshake signals between the PL330\_DMAC controller and other peripherals in the system.

Depending on the PL330\_DMAC configuration, the PL330\_RES\_CLR signal may not be used.

PL330\_DMAC\_DmaPortProtocol provides the following behaviors:

### **request()**

```
slave behavior request(uint32_t request) : void;
```

Requests from the external peripheral to the DMA controller. These are level-sensitive and are sampled by the DMA controller at specific points during the handshake. See the PL330\_DMAC.lisa implementation for more details.

### **response()**

```
master behavior response(uint32_t response) : void;
```

Responses from the DMA controller to the external component. These are transient.

## 2.46 PMUEvent protocol

Defined in \$PVLIB\_HOME/LISA/CCIRegisters.lisa.

PMUEvent provides the following behaviors:

### **fire()**

```
peer behavior fire() : void;
```

Trigger a PMU event.

## 2.47 PS2Data protocol

Defined in \$PVLIB\_HOME/LISA/PS2DataProtocol.lisa.

### **About PS2Data protocol**

This protocol is for communication between the Keyboard/Mouse Interface (KMI) and a PS/2-like device.

For efficiency, the interface is a parallel byte interface rather than a serial clock/data interface.

PS2Data provides the following behaviors:

### **getData()**

```
slave behavior getData () : uint8_t;
```

Used by the PS/2 device to get command data from the KMI.

### **putData()**

```
slave behavior putData (uint8_t data) : void;
```

Used by the PS/2 device to send device data to the KMI.

### **setClockData()**

```
master behavior setClockData (enum ps2clockdata_state) : void;
```

Used by the KMI to simulate forcing the state of the data/clock lines, to indicate whether it:

- Is able to receive data
- Wants to send a command
- Is inhibiting communication

## 2.48 PVBus protocol

Defined in \$PVLIB\_HOME/LISA/PVBusProtocol.lisa.

### About PVBus protocol

PVBus is used to provide bus connections for PV core models, or for any user-defined bus masters, to a tree of bus decoders and bus slave devices.

The bus protocol is designed to allow efficient calling through the bus decode tree, but it also implements back doors that allow bus masters to cache the decode results and access devices directly.

A bus slave component must instantiate a PVBusSlave subcomponent to provide an end point for the bus. The PVBusSlave component encapsulates all the complexity of handling the internal PVBus protocol. The PVBusSlave can be configured to handle all incoming transactions, see the example below, or as a bridge to the public PVDevice protocol.

Example of using PVBus for efficient access to memory-like storage:

```
component MemorySlave      // A component containing 64 MB of fast RAM
{
    slave port<PVBus> pvbus_s;

    master port <PVBusSlaveControl> bus_slave_control;

    composition
    {
        bus_slave : PVBusSlave(size = 0x04000000);
    }
    connection
    {
        self.pvbus_s => bus_slave.pvbus_s;
        self.bus_slave_control => bus_slave.control;
    }
}
```

```

    }
    behavior init()
    {
        bus_slave_control.setAccess(0, 0x04000000, pv::ACCESSTYPE_RW,
pv::ACCESSMODE_MEMORY);
        composition.init();
    }
}

```



The following behaviors described as internal are implemented internally by PVBUSMaster and PVBUSSlave. Devices should not implement them.

PVBUS provides the following behaviors:

### **aceSnoopRequest ()**

```
optional master behavior aceSnoopRequest( ACE::SnoopRequest* ) : void;
```

Internal behavior to support a coherency request from downstream.

### **busMapChanged ()**

```
optional master behavior busMapChanged(pv::bus_addr_t base, pv::bus_addr_t size) :
void;
```

Internal behavior used to handle cached bus decodings.

### **debugACESnoopRequest ()**

```
optional master behavior debugACESnoopRequest( ACE::SnoopRequest* ) : void;
```

Internal behavior to support a coherency request from downstream.

### **debugRead ()**

```
slave behavior debugRead(pv::ReadTransaction tx) : pv::Tx_Result;
```

Device access behavior for the PVBUS protocol.



Use of this behavior is deprecated. Use a PVBUSMaster and PVBUSSlave to send and receive PVBUS transactions.

### **debugWrite ()**

```
slave behavior debugWrite(pv::WriteTransaction tx) : pv::Tx_Result;
```

Device access behavior for the PVBUS protocol.



Note

Use of this behavior is deprecated. Use a PVBUSMaster and PVBUSSlave to send and receive PVBUS transactions.

### **discoverDownstreamChildDVMNodes ()**

```
optional slave behavior discoverDownstreamChildDVMNodes(DVM::DownstreamVisitor *) :
void;
```

Internal behavior to support DVM message passing.

Allow a PVBUS master to probe a bus port for any slaves that can propagate DVM messages.

Bus routing fabric should forward the discovery request to all slaves.

### **discoverUpstreamParentDVMNodes ()**

```
optional master behavior discoverUpstreamParentDVMNodes(DVM::UpstreamVisitor *) :
void;
```

Internal behavior to allow a PVBUS slave to probe a bus port for any masters that can respond to DVM messages.

### **doReadAccess ()**

```
optional slave behavior doReadAccess (pv::ReadRequest *) : pv::Tx_Result;
```

Internal behavior to support PVBUS re-entrant channels.

### **doWriteAccess ()**

```
optional slave behavior doWriteAccess (pv::WriteRequest *) : pv::Tx_Result;
```

Internal behavior to support PVBUS re-entrant channels.

### **read ()**

```
slave behavior read(pv::ReadTransaction tx) : pv::Tx_Result;
```

Device access behavior for the PVBUS protocol.



Note

Use of this behavior is deprecated. Use a PVBUSMaster and PVBUSSlave to send and receive PVBUS transactions.

**write()**

```
slave behavior write(pv::WriteTransaction tx) : pv::Tx_Result;
```

Device access behavior for the PVBUS protocol.



Use of this behavior is deprecated. Use a PVBUSMaster and PVBUSSlave to send and receive PVBUS transactions.

## 2.49 PVBUS2PCI2PCIDeviceProtocol protocol

Defined in \$PVLIB\_HOME/LISA/PVBUS2PCI2PCIDeviceProtocol.lisa.

### About PVBUS2PCI2PCIDeviceProtocol protocol

This is the protocol between the PVBUS2PCI and PCIDevice components.

It is used to aggregate the many connections between the two.

PVBUS2PCI2PCIDeviceProtocol provides the following behaviors:

#### **get\_device\_assignment\_info()**

```
optional slave behavior get_device_assignment_info(sg::device_assignment_info_t&
da_info_) : void;
```

#### **get\_port\_info()**

```
slave behavior get_port_info(sg::port_info_t& out_) : void;
```

PCIDevice fills out\_ with its information.

#### **get\_selective\_reg\_block\_info()**

```
optional slave behavior get_selective_reg_block_info() :
sg::ide::selective_stream_ide_reg_block_info_t;
```

A method to get the Selective Stream register block information. The information is used at the rootport to check the transactions attributes with respect to rootport DA conditions.

#### **get\_send\_error\_to\_rcec\_if()**

```
optional slave behavior get_send_error_to_rcec_if(
std::vector<uint32_t>& rciep_device_function_table,
uint32_t& bdf) : pcie::send_error_to_rcec_if*;
```

A method to check whether a downstream device is an RCEC device or not.



If so, get the pointer to `pvbus2pci` to RCEC if, which can be used to route error messages received in `PVBus2PCI/RC` towards RCECs.

It also captures RCEC associated RCiEP's device-function information to check whether the device-function passed to RCEC points to a valid RCiEP.

If not implemented or returns `nullptr` then the device is not an RCEC.



This is called during the reset phase, so the `endPoint` must cope if its own reset phase has not yet been called.

The first argument indicates the vector of `device_function_info` of RCEC associated RCiEPs. The second argument is the BDF of the downstream RCEC device. This is an optional behaviour.

### **respond\_if\_address\_is\_captured()**

```
optional master behavior respond_if_address_is_captured(pv::bus_addr_t address) :
    bool;
```

A method implemented in bridge-type devices to query whether a given address will be routed downstream to it.

### **set\_bus\_properties()**

```
optional slave behavior set_bus_properties(sg::pcie_bus_properties_t&) : void;
```

The properties of the bus can change dynamically and multiple calls to this behaviour should be expected.

## 2.50 PVBusBridgeControl protocol

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

### **About PVBusBridgeControl protocol**

Allow a component to control its `PVBusBridge` subcomponent.

`PVBusBridgeControl` provides the following behaviors:

### **configure()**

```
slave behavior configure(pv::slave_config_t*) : void;
```

Allow configuration of the transactions that are accepted by a `PVBusBridge`. By default the bridge accepts read and write transactions.

The `slave_config_t` class provides the following methods to extend the set of accepted transactions:

- `acceptACE_CleanShared_CleanInvalid_MakeInvalid()`
- `acceptACE_CleanUnique_MakeUnique()`
- `acceptEvict()`
- `acceptMemoryBarriers()`
- `acceptPrefetchOnly()`
- `acceptExclusiveTransactions()`

### **revokePrefetch()**

```
slave behavior revokePrefetch(pv::bus_addr_t base, pv::bus_addr_t top) : void;
```

Invalidates a DMI access range. `base` and `top` are included in the range.

## 2.51 PVBusCacheControl protocol

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

### About PVBusCacheControl protocol

This protocol defines behaviors that are private, subject to change, and should not be used outside of the PVBusCache component.

PVBusCacheControl provides the following behaviors:

### **createTransactionGenerator()**

```
slave behavior createTransactionGenerator(unsigned output_port) :  
    pv::TransactionGenerator*;
```

Get a transaction generator on the given output port.

### **getLineContentsForRead()**

```
slave behavior getLineContentsForRead(unsigned line_index) : const char*;
```

Get temporary read access to the line data managed by `pvBusCache`. The line must have already been initialised by calling `getLineContentsForWrite`.

### **getLineContentsForWrite()**

```
slave behavior getLineContentsForWrite(unsigned line_index) : char*;
```

Get temporary write access to the line data managed by `pvBusCache`. Allocates new storage for lines as needed.

**invalidateLineHit()**

```
slave behavior invalidateLineHit(unsigned hit_line_index,  
                                pv::CacheRevocation revoke_type) : void;
```

Revoke a line that has been marked as hitting.

**passThroughRead()**

```
slave behavior passThroughRead(unsigned output_port,  
                                pv::ReadTransaction tx) : pv::Tx_Result;
```

Pass through an unmodified read request. If a burst transaction spans more than one line, this only handles one line's worth of the burst.

**passThroughWrite()**

```
slave behavior passThroughWrite(unsigned output_port,  
                                pv::WriteTransaction tx) : pv::Tx_Result;
```

Pass through an unmodified write request. If a burst transaction spans more than one line, this only handles one line's worth of the burst.

**readFromLine()**

```
slave behavior readFromLine(pv::ReadTransaction tx,  
                             unsigned hit_line_index) : pv::Tx_Result;
```

Mark the current read transaction as hitting a cache line. All future transactions with the same attributes may be handled efficiently by `PVBusCache`, rather than being sent to the device `cacheRead()` or `cacheWrite()` handlers.

**revokeRoutingDecisions()**

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

Revoke all responses given by the `routeTransaction()` callback.

**setTimingAnnotationConfig()**

```
slave behavior setTimingAnnotationConfig(pv::PVBusCacheTAConfig cfg) : void;
```

Set the timing annotation parameters.

**writeToLine()**

```
slave behavior writeToLine(pv::WriteTransaction tx,  
                           unsigned hit_line_index) : pv::Tx_Result;
```

Mark the current write transaction as hitting a cache line. All future transactions with the same attributes may be handled efficiently by `PVBusCache`, rather than being sent to the device `cacheRead()` or `cacheWrite()` handlers.

### **writeToLineAndPassThrough()**

```
slave behavior writeToLineAndPassThrough(pv::WriteTransaction tx,
                                         unsigned hit_line_index,
                                         unsigned output_port) : pv::Tx_Result;
```

Write the transaction data into a cache line, but also pass it through to a slave port. If a burst transaction spans more than one line, this only handles one line's worth of the burst.

## 2.52 PVBusCacheDevice protocol

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

### About PVBusCacheDevice protocol

This protocol defines behaviors that are private, subject to change, and should not be used outside of the `PVBusCache` component.

`PVBusCacheDevice` provides the following behaviors:

#### **cacheRead()**

```
slave behavior cacheRead(unsigned in_port,
                           pv::ReadTransaction tx) : pv::Tx_Result;
```

Handle a read request to the cache. For burst transactions, the cache can return after handling one line's worth of transaction data, and it is called back for the first beat on the next cache line.

#### **cacheWrite()**

```
slave behavior cacheWrite(unsigned in_port,
                             pv::WriteTransaction tx) : pv::Tx_Result;
```

Handle a write request to the cache.

#### **routeTransaction()**

```
slave behavior routeTransaction(unsigned in_port,
                                  pv::Transaction tx) : pv::CacheRoutingDecision;
```

Determine whether this transaction is cacheable. If not, decide which output port should forward the transaction.

## 2.53 PVBusMapperControl protocol

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

### About PVBusMapperControl protocol

Control protocol for use with `PVBusMapper` and `PVBusModifier`.

`PVBusMapperControl` provides the following behaviors:

#### **allBusMapChanging()**

```
master behavior allBusMapChanging() : void;
```

Something connected to the control port can generate an event to the upstream that indicates the bus map is changing and asks for all requests to be remapped again.

#### **getDVMNodesCanSendTo()**

```
master behavior getDVMNodesCanSendTo (
    std::vector<pv::DVMNodeRecord>& upstream_nodes_,
    std::vector<pv::DVMNodeRecord>& downstream_nodes_
) : bool;
```

To send DVM messages, you must have a description of where to send them.

You can ask `PVBusMapper()` to give you a vector of records containing all upstream and downstream nodes. This records set is only available after first reset. You pass in a vector that you want to be filled with the appropriate nodes.

It returns true if the lists are valid, even if `empty()`, otherwise it returns false and you should try again later. When the lists become valid, the expectation is that they remain valid and there is no need to call it again.

The `PVBusMapper` will always discover upstream and downstream DVM nodes. However, there may be a logical inconsistency if you use these records to send DVM messages if you are not handling DVM messages being send to you.

You may send a `DVMMessage` using the records returned.

It is expected that if you receive a DVM message and are forwarding it to other DVM nodes, you must take care not to forward it to the DVM node that gave you it. For this purpose, use the `getPortIndex()` and `getArcWithinPort()` methods and compare the results to the `port_index_` and `arc_within_port_` given to you by the `handle*DVMMessage()` calls.

#### **getMyArcIdentifier()**

```
master behavior getMyArcIdentifier() : void*;
```

Return the arc identifier, `arc_within_port_`, that will be seen by a DVM node if we send a DVM message from this node.

**handleDownstreamDVMMessageFromUpstream()**

```
optional slave behavior handleDownstreamDVMMessageFromUpstream(
    unsigned        upstream_port_index_,
    void*           arc_within_port_,
    DVM::Message*   message_
) : DVM::error_response_t;
```

If parameter `handling_of_dvm_messages_from_upstream` is set to `handle`, this behaviour is called when a DVM message from upstream is received.

You are given the `port_index_` that the DVM message came from and an opaque pointer to the upstream master within that connection. This pointer is the same one returned in the `DVMNodeRecord` obtained from `getDVMNodesCanSendTo()`.



Do not alter `message_` and forward it. You must first copy it and then forward the copy. The message might be in use by multiple components so altering it will also alter their version.



This is a message received from upstream, so it is a downstream DVM message.

**handleSnoopRequest()**

```
optional slave behavior handleSnoopRequest(ACE::SnoopRequest* req_, bool debug_) :
    void;
```

Handle snoop requests. The `*SnoopRequest()` control port behaviors allow a `PVBusMapper` to act as an intermediary for snoop transactions on the bus.

If the `handling_of_upstream_snoop_requests` parameter is set to `handle`, this behavior is called when snoop transactions from any downstream port are received.



The snoop transaction is not automatically forwarded upstream but can be sent upstream using the `injectSnoopRequest()` behavior.

**handleUpstreamDVMMessageFromDownstream()**

```
optional slave behavior handleUpstreamDVMMessageFromDownstream(
    unsigned        downstream_port_index_,
    void*           arc_within_port_,
    DVM::Message*   message_
) : DVM::error_response_t;
```

If parameter `handling_of_dvm_messages_from_downstream` is set to `handle`, this behaviour is called when a DVM message from downstream is received.

You are given the `port_index_` that the DVM message came from and an opaque pointer to the upstream master within that connection. This pointer is the same one returned in the `DVMNodeRecord` obtained from `getDVMNodesCanSendTo()`.



Note

Do not alter `message_` and forward it. You must first copy it and then forward the copy. The message might be in use by multiple components so altering it will also alter their version.



Note

This is a message received from downstream and so it is an upstream DVM message.

### **injectSnoopRequest()**

```
master behavior      injectSnoopRequest(ACE::SnoopRequest* req_, bool debug_) :
void;
```

Issue a snoop transaction upstream.

### **printDVMNodes()**

```
master behavior printDVMNodes(std::ostream&, const std::string& indent_) : void;
```

Print to the stream a text description of the nodes that it has currently found.

### **remap()**

```
slave behavior remap(
    pv::RemapRequest& req_
) : unsigned;
```

Return the port that this transaction should be filtered to, based on the attributes and the address information held in the `RemapRequest` object. You may also indicate a remapping of the attributes and address in this call.

You may tag this decision with zero, one, or more objects of a type derived from `RemapDecisionGroup`. This allows you to revoke all decisions tagged with the same `RemapDecisionGroup` object. This object is allocated and owned by the component implementing the `remap()` function. See the `RemapDecisionGroup` class for more details.

The remapper must be consistent with respect to its decisions and so they must be statically determined.

The return value is a port number of `pbus_m`, or either of the special values:

**PVBUSMAPPER\_ABORT**

Abort all accesses

**PVBUSMAPPER\_IGNORE**

Treat all accesses as Read-As-Zero, Writes Ignored (**RAZ/WI**)

Any other value is considered an error.

**reset ()**

```
master behavior reset() : void;
```

Signal a reset of the bus mapper bus interfaces. This is equivalent to an assert of the reset signal.

**sendAllBusMapChangingToUpstreamPort ()**

```
master behavior sendAllBusMapChangingToUpstreamPort (
    unsigned upstream_port_index_
) : void;
```

Something connected to the control port can generate an event to an upstream port that indicates a bus map range is changing and asks for matching requests to be remapped again.

## 2.54 PVBUSOverTLMControl protocol

Defined in \$PVLIB\_HOME/examples/SystemCExport/Bridges/PVBus2AMBAPVACE.lisa.

### About PVBUSOverTLMControl protocol

This version of the PVBus to AMBA-PV bridge enables you to pass back all the coherency information from ACP.



This version makes use of a private and undocumented API that is not intended to be supported and will change in future releases.

PVBusOverTLMControl provides the following behaviors:

**routeAccesses ()**

```
slave behavior routeAccesses (BUS_RouteOption destination) : bool;
```



## 2.55 PVBUSRouterControl protocol

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

### About PVBUSRouterControl protocol

Allow the construction of arbitrary routing decisions.

`PVBUSRouterControl` provides the following behaviors:

#### **filter()**

```

slave behavior filter(
    const pv::TransactionAttributes* attributes_,
    pv::bus_addr_t                  page_base_,
    bool                            is_read_
) : unsigned;

```

Return the port that this transaction should be filtered to, based on the attributes and the `page_base_`, which is the address aligned to 4 KiB.

The filter must be consistent with respect to its filtering decisions and so they must be statically determined.

The return value is a port number of `pvbus_m`, or either of the special values:

#### **PVBUSROUTER\_ABORT**

Abort all accesses

#### **PVBUSROUTER\_IGNORE**

Treat all accesses as Read-As-Zero, Writes Ignored (**RAZ/WI**)

Any other value is considered an error.

## 2.56 PVBUSSlaveControl protocol

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

### About PVBUSSlaveControl

Allow a component to configure its `PVBUSSlave` subcomponent.

This gives it control over mapping regions of device memory to be RAM, ROM, or device memory.

The `PVBUSSlave` automatically routes incoming bus accesses according to this configuration. Accesses to device memory, or writes to ROM memory, are routed to the device port, which the component should use to provide implementations of the `read()` and `write()` behaviors.

`PVBUSSlaveControl` provides the following behaviors:

**closeRegionIterHandle()**

```
slave behavior closeRegionIterHandle(uint32_t iter_handle) : void;
```

A caller may close an iterator opened by `getRegionIterHandle()` at any time using `closeRegionIterHandle()`. This deallocates the iterator and further uses of the handle are invalid.

**configure()**

```
slave behavior configure(pv::slave_config_t*) : void;
```

Allow configuration of the transactions that are accepted by a PVBUSlave. By default the slave accepts read and write transactions.

The `slave_config_t` class provides the following methods to extend the set of accepted transactions:

- `acceptACE_CleanShared_CleanInvalid_MakeInvalid()`
- `acceptACE_CleanUnique_MakeUnique()`
- `acceptEvict()`
- `acceptMemoryBarriers()`
- `acceptPrefetchOnly()`
- `acceptExclusiveTransactions()`

**getNextRegionInfo()**

```
slave behavior getNextRegionInfo(uint32_t iter_handle,
                                pv::PVBUSlaveRegionInfo *info) : bool;
```

After calling `getRegionIterHandle()`, the caller may repeatedly call `getNextRegionInfo()` with the provided `iter_handle`. If a region is found, the behavior returns true and the `info` struct is written to if the pointer is non-null. The region's data may be accessed using `getReadStorage()` or `getWriteStorage()`.

Regions may be returned in any order, and may be of any size or alignment, but no two regions overlap.

An implementation may decide not to report regions that have been allocated, but filled entirely with the default fill pattern, or regions allocated, but containing only the data they had at simulation start.

On reaching the last region, the iterator is automatically closed. If the handle is invalid or there are no further regions, the behavior returns false.

**getReadStorage()**

```
slave behavior getReadStorage(pv::bus_addr_t address,
                             pv::bus_addr_t *limit) : const uint8_t*;
```

Get read access to the underlying memory storage provided by the PVBUSlave. The parameters are:

**address**

Byte address to request access to.

**limit**

Returns the address limit for the contiguous region.

The returned pointer can be used to directly access all memory locations from `address` to `limit-1`. The returned pointer is only guaranteed to remain valid until the next bus access or simulation cycle.

Modifying memory using `getWriteStorage()` does not inform any of the global exclusive monitors of the update.

### **getRegionIterHandle()**

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

An iterator-like API that allows a PVBUSlave that provides storage to report all the regions of the address space that have backing store.

The iteration begins by calling `getRegionIterHandle()`. This allocates an iterator and if successful, returns a non-zero `iter_handle` to identify it.

### **getWriteStorage()**

```
slave behavior getWriteStorage(pv::bus_addr_t address,  
                               pv::bus_addr_t *limit) : uint8_t*;
```

Get write access to the underlying memory storage provided by the PVBUSlave. The parameters are:

**address**

Byte address to request access to.

**limit**

Returns the address limit for the contiguous region.

The returned pointer can be used to directly access all memory locations from `address` to `limit-1`. The returned pointer is only guaranteed to remain valid until the next bus access or simulation cycle.

Modifying memory using `getWriteStorage()` does not inform any of the global exclusive monitors of the update.

### **provideReadStorage()**

```
optional slave behavior provideReadStorage(pv::bus_addr_t device_base,  
                                           pv::bus_addr_t device_limit,  
                                           const uint8_t *storage) : void;
```

Allows a device to provide its own memory region to implement memory storage. For example, a device may want to ensure that the underlying memory is implemented in one contiguous region, or is allocated from a special region, for example video memory or memory-mapped memory.

The caller must allow the PVBUSSlave to take ownership of all accesses to this memory. In other words, the caller must call `getWriteStorage()` before modifying the contents of this memory region.

`device_base` and `device_limit` must be 4 KB-aligned.

Read latency for the range, which is used when Timing Annotation is enabled, is set from the PVBUSSlave `read_latency` parameter.

### **provideReadStorageEx()**

```
slave behavior provideReadStorageEx(pv::bus_addr_t device_base,  
                                   pv::bus_addr_t device_limit,  
                                   const uint8_t *storage,  
                                   double read_latency) : void;
```

This behavior is the same as `provideReadStorage()` but with an additional parameter to specify an average latency per byte, which is used when Timing Annotation is enabled.

### **provideReadWriteStorage()**

```
optional slave behavior provideReadWriteStorage(  
    pv::bus_addr_t device_base,  
    pv::bus_addr_t device_limit,  
    uint8_t *storage) : void;
```

Allows a device to provide its own memory region to implement memory storage. For example, a device may want to ensure that the underlying memory is implemented in one contiguous region, or is allocated from a special region, for example video memory or memory-mapped memory.

The caller must allow the PVBUSSlave to take ownership of all accesses to this memory. In other words, the caller must call `getWriteStorage()` before modifying the contents of this memory region.

`device_base` and `device_limit` must be 4 KB-aligned.

Read and write latencies for the range, which are used when Timing Annotation is enabled, are set from the PVBUSSlave `read_latency` and `write_latency` parameters.

### **provideReadWriteStorageEx()**

```
slave behavior provideReadWriteStorageEx(  
    pv::bus_addr_t device_base,  
    pv::bus_addr_t device_limit,  
    uint8_t *storage,  
    double read_latency,  
    double write_latency) : void;
```

This behavior is the same as `provideReadWriteStorage()` but with an additional parameter to specify an average latency per byte, which is used when Timing Annotation is enabled.

### **provideWriteStorage()**

```
optional slave behavior provideWriteStorage(pv::bus_addr_t device_base,
                                           pv::bus_addr_t device_limit,
                                           uint8_t*storage) : void;
```

Allows a device to provide its own memory region to implement memory storage. For example, a device may want to ensure that the underlying memory is implemented in one contiguous region, or is allocated from a special region, for example video memory or memory-mapped memory.

The caller must allow the PVBUSSlave to take ownership of all accesses to this memory. In other words, the caller must call `getWriteStorage()` before modifying the contents of this memory region.

`device_base` and `device_limit` must be 4 KB-aligned.

Write latency for the range, which is used when Timing Annotation is enabled, is set from the PVBUSSlave `write_latency` parameter.

### **provideWriteStorageEx()**

```
slave behavior provideWriteStorageEx(pv::bus_addr_t device_base,
                                     pv::bus_addr_t device_limit,
                                     uint8_t*storage,
                                     double write_latency) : void;
```

This behavior is the same as `provideWriteStorage()` but with an additional parameter to specify an average latency per byte, which is used when Timing Annotation is enabled.

### **reset()**

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

Signal a reset of the bus slave interface.

This is equivalent to an assert of the reset signal.

### **setAccess()**

```
slave behavior setAccess(pv::bus_addr_t base,
                         pv::bus_addr_t top,
                         pv::accessType type,
                         pv::accessMode mode) : void;
```

Define how accesses are routed for a given range of device addresses. The parameters are:

#### **base**

Start address of the range to be configured, 4 KB-aligned.

**top**

End address, 4 KB-aligned.

**type**

Type of access to configure. Possible values:

- `ACCESSTYPE_READ`
- `ACCESSTYPE_WRITE`
- `ACCESSTYPE_RW`

**mode**

The new mode for accesses.

The following access modes control how to treat accesses of the selected type, within the chosen range:

**ACCESSMODE\_MEMORY**

Access data storage, which is managed by the PVBusSlave.

**ACCESSMODE\_DEVICE**

Route request to the device port on the slave.

**ACCESSMODE\_ABORT**

Generate an abort on the transaction.

**ACCESSMODE\_IGNORE**

Ignore the transaction. Reads return 0.

**setFillPattern()**

```
slave behavior setFillPattern (uint32_t fill1, uint32_t fill2) : void;
```

Set the default fill pattern for RAM or ROM regions. This should be called before any memory accesses occur, and allows memory to be prefilled with an alternating two-word pattern.

## 2.57 PVC2C protocol

Defined in `$PVLIB_HOME/LISA/PVC2CProtocol.lisa`.

### About PVC2C protocol

PVC2C protocol models chip-to-chip connections. The data transfer in PVC2C is unidirectional, which is similar to CXS. Therefore, there is a dedicated interface for each transmitter and receiver.

`pvc2c` provides the following behaviors:

**discover\_pvc2c\_chips()**

```
slave behavior discover_pvc2c_chips(pvc2c::pvc2c_discovery_req_t*) : void;
```

**get\_haid()**

```
slave behavior get_haid() : pvc2c::haid_t;
```

**get\_pvc2c\_properties()**

```
slave behavior get_pvc2c_properties() : pvc2c::pvc2c_port_properties_t;
```

## 2.58 PVCacheDebugRam protocol

Defined in \$PVLIB\_HOME/LISA/PVCache\_DebugRamProtocol.lisa.

PVCacheDebugRam provides the following behaviors:

**getAttribute()**

```
slave behavior getAttribute(pv::PVCache_DebugRamPort::Attribute attribute, unsigned index) : uint64_t;
```

Get an attribute value from the cache.

This interface exposes certain numeric attributes of the cache. The parameters are:

**attribute**

Selects the attribute value to return. Must be one of the A\_\* enum constants specified in PVCache\_DebugRamPort.h.

**index**

Currently unused and must be set to 0. It is intended to expose arrays of attributes, for example per-set/way attributes.

This behavior returns a numeric attribute value. It returns 0 for unknown or unsupported attributes or for index out of range.

**getConfig()**

```
slave behavior getConfig(pv::PVCache_DebugConfig& config) : void;
```

Obtain cache configuration, mainly geometry for now.

**getDeferredActions()**

```
slave behavior getDeferredActions() : sg_deferred_actions::deferred_actions_t*;
```

Get a handle to the deferred\_actions\_t object used by the implementation of the cache system.

The cache system has the ability to defer internal events until a re-entrant safe point by pushing them onto a deferred\_actions\_t object.

This method gets a handle to the object for validation purposes.

### **peekLine()**

```
slave behavior peekLine(const pv::PVCache_DebugFilter& filter,  
pv::PVCache_DebugLine& buffer) : bool;
```

Peek the cache line location, tag, and content. The parameters are:

#### **filter**

Reference to a cache lookup filter. Filters can be constructed for lookup by address, index, set/way, and so on.

#### **buffer**

Reference to a buffer for the returned cache line location, tag, and content. If `buffer` is constructed with a zero length cache line, the cache line content is not returned.

If cache lookup is by address and fails to hit, the `location` member of `buffer` is updated with the first way in the cache where the cache line could have been.

This behavior returns the cache lookup/hit status. If false, the tag and the content in the buffer are not updated.

For lookup by index or set/way, if the selected line is within the boundary of the cache, the routine returns true and the `tag.valid` data member in `buffer` indicates whether the cache line, tag, and content members contain valid data.

For lookup by address, the return status indicates a hit or miss in the cache. A miss does not cause any further transactions downstream or allocation into the cache.

### **pokeLine()**

```
slave behavior pokeLine(const pv::PVCache_DebugFilter& filter, const  
pv::PVCache_DebugLine& buffer) : bool;
```

Poke the cache line tag and content data.

The cache may not be able to accept all cache line tag modifications and may ignore some or all such modifications. But overwriting cache line content is always supported by the cache.

The parameters are:

#### **filter**

Reference to a cache lookup filter. Filters can be constructed for lookup by address, index, set/way and so on.

#### **buffer**

Reference to a buffer with cache line tag and content data to set. The cache line location member in `buffer` is ignored. If `buffer` is constructed with a zero length cache line, the cache line content is not set.



This behavior returns the cache lookup/hit status. For lookup by index or set/way, if the selected line is within the boundary of the cache, the routine returns true and the cache line state data inside the cache is modified.

For lookup by address, the return status indicates a hit or miss in the cache. A miss does not cause any further transactions downstream, allocation into the cache, or modification of cache line state.

## 2.59 PVCacheMaintenance protocol

Defined in \$PVLIB\_HOME/LISA/PVCacheMaintenance.lisa.

PVCacheMaintenance provides the following behaviors:

### cacheSizeOverride()

```
slave behavior cacheSizeOverride( unsigned cache_line_size_in_bytes_,
                                   unsigned number_of_sets_,
                                   unsigned number_of_ways_ ) : void;
```

Private internal functionality of the cache implementation. Do not use.

Override the cache size. All the contents of the cache will be lost at this point. If any lines have been allocated then ACE state may be corrupted across the system. The caller is responsible for ensuring that no transactions are in flight when calling this.

### clean\_all()

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

Clean the entire cache, flushing all dirty lines.

### clean\_and\_invalidate\_all()

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

Clean and invalidate the entire cache, evicting all lines without cleaning.

### clean\_and\_invalidate\_by\_addr()

```
optional slave behavior clean_and_invalidate_by_addr(
    pv::bus_addr_t addr, bool is_non_secure) : void;
```

Clean and invalidate by PA. This should be used in preference to `clean_by_addr()` followed by `invalidate_by_addr()` as a write could occur between the two and then the `invalidate_by_addr()` would invalidate dirty data.

### clean\_and\_invalidate\_by\_addr\_by\_pas()

```
optional slave behavior clean_and_invalidate_by_addr_by_pas(
```

```
pv::bus_addr_t addr, pv::PASpace_t pas) : void;
```

### **clean\_and\_invalidate\_by\_set\_way()**

```
optional slave behavior clean_and_invalidate_by_set_way(
    uint32_t set, uint32_t way, bool is_non_secure) : void;
```

Clean and invalidate by set/way. Secure evicts any, non-secure only evicts non-secure entries.

### **clean\_and\_invalidate\_by\_set\_way\_by\_pas()**

```
optional slave behavior clean_and_invalidate_by_set_way_by_pas(
    uint32_t set, uint32_t way, pv::PASpace_t pas) : void;
```

### **clean\_by\_addr()**

```
slave behavior clean_by_addr(pv::bus_addr_t addr, bool is_non_secure) : void;
```

Clean by PA, evicting the lines that match.

### **clean\_by\_addr\_by\_pas()**

```
optional slave behavior clean_by_addr_by_pas(
    pv::bus_addr_t addr, pv::PASpace_t pas) : void;
```

### **clean\_by\_set\_way()**

```
slave behavior clean_by_set_way(uint32_t set, uint32_t way, bool is_non_secure) :
    void;
```

Clean by set/way. Secure evicts any, non-secure only evicts non-secure entries.

### **clean\_by\_set\_way\_by\_pas()**

```
optional slave behavior clean_by_set_way_by_pas(
    uint32_t set, uint32_t way, pv::PASpace_t pas) : void;
```

### **enableLocalDVMMessageProcessing()**

```
slave behavior enableLocalDVMMessageProcessing( bool on_ ) : void;
```

Enable or disable whether the current cache handles DVM messages locally.

When the cache was created, there may have been the option of telling it to startup to ignore\_local\_dvm\_messages or not. This now makes that a dynamic behaviour.

### **enableUpstreamAcceptsDVM()**

```
slave behavior enableUpstreamAcceptsDVM( unsigned upstream_port_, bool on_ ) :
    void;
```

Enable or disable which of the upstream ports are currently accepting DVM snoop requests.

This is used to override the local cache behavior.

### **enableUpstreamAcceptsSnoopRequests()**

```
slave behavior enableUpstreamAcceptsSnoopRequests( unsigned upstream_port_, bool
on_ ) : void;
```

Enable or disable which of the upstream ports are currently accepting ACE snoop requests.

This is used to override the local cache behavior.

### **find\_in\_cache()**

```
slave behavior find_in_cache(
    const pv::MemoryAttributes &memory_attributes_,
    pv::bus_addr_t address_
) : bool;
```

Test whether this layer of cache contains a given line.



The security world is encoded in the `memory_attributes_` parameter. The exclusive, cache maintenance, and debug flags are ignored.

### **getCacheStateModelled()**

```
slave behavior getCacheStateModelled() : bool;
```

Get the current value of “cache state modelled”.

### **getEnabled()**

```
slave behavior getEnabled(bool is_non_secure) : bool;
```

Get the enabled state.

### **getLockDown()**

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

Get a bit array controlling which cache ways are locked down.

### **invalidate\_all()**

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

Invalidate the entire cache, evicting all lines without cleaning.

**invalidate\_by\_addr()**

```
slave behavior invalidate_by_addr(pv::bus_addr_t addr, bool is_non_secure) : void;
```

Invalidate by PA, evicting the lines that match.

**invalidate\_by\_addr\_by\_pas()**

```
optional slave behavior invalidate_by_addr_by_pas(
    pv::bus_addr_t addr, pv::PASpace_t pas) : void;
```

**invalidate\_by\_set\_way()**

```
slave behavior invalidate_by_set_way(uint32_t set, uint32_t way, bool
    is_non_secure) : void;
```

Invalidate by set/way. Secure evicts any, non-secure only evicts non-secure entries.

**invalidate\_by\_set\_way\_by\_pas()**

```
optional slave behavior invalidate_by_set_way_by_pas(
    uint32_t set, uint32_t way, pv::PASpace_t pas) : void;
```

**preload()**

```
slave behavior preload(
    const pv::MemoryAttributes &memory_attributes_,
    pv::bus_addr_t address_,
    bool make_unique_,
    sg::ticks_t& local_time_
) : pv::Tx_Result;
```

Preload a line into this layer of the cache.

If you ask for it to be unique, it performs all the cache coherency operations to make it unique to that cache, assuming it is shared. This means that a write to that cache does not have to perform extra coherency operations, assuming it is still unique in the cache at that point. This is intended as a primitive to model preload for read and preload for write.



The security world is encoded in the `memory_attributes_` parameter. The exclusive, cache maintenance, and debug flags are ignored.

**setBitmapOfDownstreamPortsThatIsDomainBoundaryForReallyNonShared()**

```
optional slave behavior
    setBitmapOfDownstreamPortsThatIsDomainBoundaryForReallyNonShared(uint64_t) : void;
```

Override the

`bitmap_of_downstream_ports_that_is_the_domain_boundary_for_really_non_shared` parameter.

This bitmap is used to indicate whether an nsh request treated as sh should be recovered back to nsh when it goes out to the downstream.

### **setBitmapOfUpstreamPortsThatTreatNonSharedAsShared()**

```
optional slave behavior setBitmapOfUpstreamPortsThatTreatNonSharedAsShared(uint64_t
    bitmap_of_upstream_ports_that_treat_nsh_as_sh) : void;
```

Override the `bitmap_of_upstream_ports_that_treat_non_shared_as_shared` parameter.

This bitmap is used to indicate that a non-shared request is treated as shared within the cluster or cache.

### **setCacheStateModelled()**

```
slave behavior setCacheStateModelled(bool modelled) : void;
```

Set the “cache state modelled” state.

### **setEnabled()**

```
slave behavior setEnabled(bool enabled, bool is_non_secure) : void;
```

Set the enabled state.

### **setIsInner()**

```
slave behavior setIsInner(bool is_inner) : void;
```

Set the domain for the cache as inner or outer.

### **setLockDown()**

```
slave behavior setLockDown(uint32_t lock) : void;
```

Set a bit array controlling which cache ways are locked down.

### **setNoDistinctionBetweenIshAndOsh()**

```
slave behavior setNoDistinctionBetweenIshAndOsh(bool
    no_distinction_between_ish_and_osh_) : void;
```

Reconfigure whether the cache treats the distinction between Inner-Shareability and Outer-Shareability as meaningful when matching attributes.



The Outer/Inner Shareability distinction is preserved on the bus.

---



The caller is responsible for ensuring that no transactions are in flight when calling this behaviour.

## 2.60 PVDevice protocol

Defined in \$PVLIB\_HOME/LISA/PVDeviceProtocol.lisa.

### About PVDevice protocol

Simple bus protocol that allows a LISA component to handle bus read/write transactions using a PVBusSlave subcomponent.

Examples of usage:

```
component SimpleSlave
{
  composition
  {
    bus_slave : PVBusSlave(size = 0x1000);
  }
  connection
  {
    self.pvbus_s => bus_slave.pvbus_s;
    bus_slave.device => self.device;
  }

  slave port <PVBus> pvbus_s;

  internal slave port<PVDevice> device
  {
    behavior read(pv::ReadTransaction tx) : pv::Tx_Result
    {
      switch(tx.getAddress() & ~3)
      {
        case 0: return tx.setReturnData32(0x12345678);
        default: return tx.generateAbort();
      }
    }
    behavior write(pv::WriteTransaction tx) : pv::Tx_Result
    {
      uint32_t data = tx.getData32();
      tx.writeComplete();
    }
    behavior debugRead(pv::ReadTransaction tx) : pv::Tx_Result
    {
      return device.read(tx);
    }
    behavior debugWrite(pv::WriteTransaction tx) : pv::Tx_Result
    {
      return device.write(tx);
    }
  }
}
```

PVDevice provides the following behaviors:

**debugRead()**

```
slave behavior debugRead(pv::ReadTransaction tx) : pv::Tx_Result;
```

Enable the device to handle a debug read transaction.

**debugWrite()**

```
slave behavior debugWrite(pv::WriteTransaction tx) : pv::Tx_Result;
```

Enable the device to handle a debug write transaction.

**read()**

```
slave behavior read(pv::ReadTransaction tx) : pv::Tx_Result;
```

Enable the device to handle a bus read transaction.

**revokePrefetch()**

```
master behavior revokePrefetch(pv::RevokeTransaction* tx,  
    pv::range_t<pv::bus_addr_t> range) : void;
```

Allow the slave to revoke any prefetch information given to the master.

This revokes both read and write prefetches for the range given. The revoke transactions can be obtained using `tx.getPayload()->getRevokeTransaction()`.

See the equivalent behaviour in `PVBus.h` for more information.

**write()**

```
slave behavior write(pv::WriteTransaction tx) : pv::Tx_Result;
```

Enable the device to handle a bus write transaction.

## 2.61 PVTransactionMaster protocol

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

### About PVTransactionMaster protocol

This protocol exists to allow bus masters to instantiate `TransactionGenerator` objects on the control port of a `PVBusMaster` subcomponent.

Any number of `TransactionGenerator` objects can be created from a single `PVBusMaster`.

They should be allocated at startup, because allocating a new `TransactionGenerator` for each transaction is expensive.

It is most efficient to have one `TransactionGenerator` for each data stream that is being accessed. For example, to get maximum efficiency from the PVBUS system, a DMA memory transfer should use one generator for the reads and one for the writes.

The `TransactionGenerator` class is defined in `$PVLIB_HOME/include/pv/PVBusMaster.h`.

`PVTransactionMaster` provides the following behaviors:

#### **`createRandomContextTransactionGenerator()`**

```
slave behavior createRandomContextTransactionGenerator() :
    pv::RandomContextTransactionGenerator*;
```

Return a new instance of a `RandomContextTransactionGenerator` object.

#### **`createStreamingTransactionGenerator()`**

```
slave behavior createStreamingTransactionGenerator() :
    pv::StreamingTransactionGenerator*;
```

Return a new instance of a `StreamingTransactionGenerator` object.

#### **`createTransactionGenerator()`**

```
slave behavior createTransactionGenerator() : pv::TransactionGenerator*;
```

Return a new instance of a `TransactionGenerator` object.

#### **`reset()`**

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

Signal a reset of the bus master interface. This is equivalent to a deassert of the reset signal.

## 2.62 PVWriteBuffer\_BarrierPort protocol

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

`PVWriteBuffer_BarrierPort` provides the following behaviors:

#### **`CleanByAddr()`**

```
slave behavior CleanByAddr(bus_addr_t addr, bool ns) : void;
```

#### **`CleanByAddrNSNSE()`**

```
slave behavior CleanByAddrNSNSE(bus_addr_t addr, bool ns, bool nse) : void;
```



**CleanByAddrPAs()**

```
slave behavior CleanByAddrPAs(bus_addr_t addr, pv::PASpace_t pas) : void;
```

**notify()**

```
slave behavior notify(PVWriteBufferComponentBarrier_t type) : void;
```

## 2.63 PVWriteBuffer\_SErrorPort protocol

Defined in \$PVLIB\_HOME/LISA/PVWriteBuffer.lisa.

PVWriteBuffer\_SErrorPort provides the following behaviors:

**notify()**

```
slave behavior notify(const Tx_Result& result,
                      bus_addr_t address,
                      const Payload& payload,
                      const uint8_t* faultp,
                      PASpace_t pas) : void;
```

## 2.64 PVWriteBuffer\_VmidBarrierPort protocol

Defined in \$PVLIB\_HOME/LISA/PVWriteBuffer.lisa.

PVWriteBuffer\_VmidBarrierPort provides the following behaviors:

**notify()**

```
slave behavior notify(PVWriteBufferComponentBarrier_t type, unsigned vmid) : void;
```

## 2.65 SC\_ClockRateControl protocol

Defined in \$PVLIB\_HOME/examples/SystemCExport/Common/Protocols/LISA/SC\_ClockRateControlProtocol.lisa.

**About SC\_ClockRateControl protocol**

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

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

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

SC\_ClockRateControl provides the following behaviors:

#### **set64\_m()**

```
master behavior set64_m(uint64_t mul, uint64_t div) : void;
```

Set clock rate. New clock rate = mul / div.

#### **set64\_s()**

```
slave behavior set64_s(uint64_t mul, uint64_t div) : void;
```

Set clock rate. New clock rate = mul / div.

#### **set\_m()**

```
master behavior set_m(uint32_t mul, uint32_t div) : void;
```

Set clock rate. New clock rate = mul / div.

#### **set\_s()**

```
slave behavior set_s(uint32_t mul, uint32_t div) : void;
```

Set clock rate. New clock rate = mul / div.

## 2.66 SC\_ClockSignal protocol

Defined in \$PVLIB\_HOME/examples/SystemCEExport/Common/Protocols/LISA/SC\_ClockSignalProtocol.lisa.

### About SC\_ClockSignal protocol

SystemC export equivalent of the LISA+ [ClockSignal protocol](#).

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

ClockSignal output ports are provided on the following library components:

#### **MasterClock**

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

## ClockDivider

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

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



Note

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

## Example system using ClockSignal

```

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

```

sc\_ClockSignal provides the following behaviors:

### current\_ticks\_m()

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

Private internal method used between Scheduler components.

### current\_ticks\_s()

```
slave behavior current_ticks_s() : uint64_t;
```

Private internal method used between Scheduler components.

### get\_clock\_m()

```
master behavior get_clock_m() : sg::FrequencySource*;
```

Private internal method used between Scheduler components.

### **get\_clock\_s()**

```
slave behavior get_clock_s() : sg::FrequencySource*;
```

Private internal method used between Scheduler components.

### **rate\_in\_hz\_m()**

```
master behavior rate_in_hz_m() : double;
```

Private internal method used between Scheduler components.

### **rate\_in\_hz\_s()**

```
slave behavior rate_in_hz_s() : double;
```

Private internal method used between Scheduler components.

### **set\_clock\_m()**

```
master behavior set_clock_m(sg::FrequencySource* _sg_frequencysource_0) : void;
```

Private internal method used between Scheduler components.

### **set\_clock\_s()**

```
slave behavior set_clock_s(sg::FrequencySource* _sg_frequencysource_0) : void;
```

Private internal method used between Scheduler components.

## 2.67 SC\_VirtualEthernet protocol

Defined in \$PVLIB\_HOME/examples/SystemCEExport/Common/Protocols/LISA/  
SC\_VirtualEthernetProtocol.lisa.

### About SC\_VirtualEthernet protocol

SystemC equivalent of [VirtualEthernet protocol](#).

SC\_VirtualEthernet provides the following behaviors:

### **send\_to\_master\_m()**

```
master behavior send_to_master_m(EthernetFrame* frame) : void;
```

**send\_to\_slave\_s()**

```
slave behavior send_to_slave_s(EthernetFrame* frame) : void;
```

## 2.68 SMMUv3AEMIdentifyProtocol protocol

Defined in \$PVLIB\_HOME/LISA/SMMUv3AEMIdentifyProtocol.lisa.

**About SMMUv3AEMIdentifyProtocol protocol**

Architecturally, a transaction comes into the SMMU model with the following side band signals:

- Security State Determination (SSD):
  - 0  
Transaction belongs to a device controlled by the secure world
  - 1  
Transaction belongs to a device controlled by the non-secure world
  - 2  
Transaction belongs to a device controlled by the root world
  - 3  
Transaction belongs to a device controlled by the realm world
- StreamID
- SubStreamID and SubStreamID valid

How these are transported in the system is SoC-dependent.

The SMMU model requires that the SoC provides a way of determining this information by providing the `identify()` behaviour.

SMMUv3AEMIdentifyProtocol provides the following behaviors:

**identify()**

```
slave behavior identify(
    unsigned                tbu_number_,
    const pv::TransactionAttributes* attributes_,
    bool*                   out_ssd_ns_,
    unsigned*               out_streamid_,
    unsigned*               out_substreamid_ // ~0u if no substreamid
) : void;
```

**identify\_2()**

```
optional slave behavior identify_2(
    unsigned                tbu_number_,
    const pv::TransactionAttributes* attributes_,
    unsigned*               out_ssd_,      // 0 -- s, 1
    -- ns, 2 -- rt, 3 -- rl
    uint64_t*               out_streamid_, // ~0ull
    if NoStreamID
```

```

        unsigned*                                out_substreamid_, // ~0u if
no substreamid
        // Added between 11.16 and 11.17
        SMMUv3AEM::smmu3aem_identify_protocol_extra_t* out_extra_ // For NoStreamID
only
    ) : void;

```

## 2.69 SchedulerInterfaceControl protocol

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

### About SchedulerInterfaceControl protocol

This protocol is used to access the Fast Models scheduler.

`SchedulerInterfaceControl` provides the following behaviors:

#### **waitTicks()**

```
slave behavior waitTicks(uint64_t ticks) : void;
```

Let the time of the calling thread advance by `ticks`, relative to `clk_in`.

## 2.70 SchedulerThreadControl protocol

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

### About SchedulerThreadControl protocol

This protocol is used to control the behavior of the `SchedulerThread` component and also to run the actual thread code.

`SchedulerThreadControl` provides the following behaviors:

#### **setupThread()**

```
slave behavior setupThread(unsigned index, void *args, const
sg::SchedulerThreadParameters *parameters) : void;
```

Set up a new thread.

This function must only be called if more than one thread should be handled through this `schedulerThread` instance. A default thread is always started with `index=0` and `args=0`.

Calling this function may or may not yield to other threads, not necessarily the newly-created thread.

Specifying `parameters = 0` has the same semantics as specifying a default constructed `SchedulerThreadParameters()`. The instance pointed to by `parameters` is not used after `setupThread()` returns.

### **threadProc()**

```
master behavior threadProc(unsigned index, void *args) : void;
```

Actual thread function. The `index` and `args` parameters are set to 0, 0 for the default thread and are specified in the `setupThread(index args)` call for all additional threads.

### **waitTicks()**

```
slave behavior waitTicks(uint64_t ticks) : void;
```

Let the time of this thread advance by `ticks`, relative to `clk_in`.

This is the same as `SchedulerInterfaceControl.waitTicks()`.

## 2.71 SchedulerThreadEventControl protocol

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

### **About SchedulerThreadEventControl protocol**

This protocol is used to control the behavior of the `ThreadSignal` component.

`SchedulerThreadEventControl` provides the following behaviors:

### **notify()**

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

Unblock any Fast Models threads waiting on this event.

Ignored if no threads are waiting. The event is not buffered until another thread tries to wait.

### **wait()**

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

Block the current Fast Models thread, for example the `schedulerThread` instance, until anything calls `notify()`.

## 2.72 SerialData protocol

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

### About SerialData protocol

This protocol is implemented as a parallel interface for efficiency. All communication is driven by the master port.

`SerialData` provides the following behaviors:

#### **dataReceive()**

```
peer behavior dataReceive() : uint16_t;
```

Used by the master to receive data from the slave.

**Table 2-3: Bits for dataReceive()**

Bits	Function
15:13	Reserved
12	Set when no data available for reading
11	Reserved
10	Break error
9:8	Reserved
7:0	Receive data

#### **dataTransmit()**

```
peer behavior dataTransmit(uint16_t data) : void;
```

Used by the master to send data to the slave.

**Table 2-4: Bits for dataTransmit()**

Bits	Function
15:8	Reserved
7:0	Transmit data

#### **signalsGet()**

```
peer behavior signalsGet() : uint8_t;
```

Used by the master to get the current signal status.

**Table 2-5: Bits for signalsGet()**

Bits	Function
7:4	Reserved



Bits	Function
3	DCD
2	DSR
1	CTS
0	RI

**signalsSet()**

```
peer behavior signalsSet(uint8_t signal) : void;
```

Used by the master to set the current signal status.

**Table 2-6: Bits for signalsSet()**

Bits	Function
7	Out1
6	Out2
5	RTS
4	DTR
3:0	Reserved

## 2.73 Signal protocol

Defined in \$PVLIB\_HOME/LISA/SignalProtocol.lisa.

**About Signal protocol**

The Signal protocol provides a single method that allows a master to set or clear a signal. This can be used for any level-sensitive signalling

The `sg::Signal::State` enumeration provides two values:

```
sg::Signal::Set
sg::Signal::Clear
```

signal provides the following behaviors:

**setValue()**

```
peer behavior setValue(sg::Signal::State) : void;
```

Set signal value. Allowed values:

- `sg::Signal::Set`
- `sg::Signal::Clear`

## 2.74 StateSignal protocol

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

### About StateSignal protocol

The StateSignal protocol provides one method that allows a master to set or clear a signal and another allowing the master to retrieve the current state from a slave. This can be used for any level-sensitive signalling

The `sg::Signal::State` enumeration provides two values:

```
sg::Signal::Set  
sg::Signal::Clear
```

`StateSignal` provides the following behaviors:

#### **getValue()**

```
peer behavior getValue() : sg::Signal::State;
```

Returns the state of the signal.

#### **setValue()**

```
peer behavior setValue(sg::Signal::State) : void;
```

Set the signal value. Allowed values:

- `sg::Signal::Set`
- `sg::Signal::Clear`

## 2.75 SystemCCoprocBusProtocol protocol

Defined in `$PVLIB_HOME/examples/SystemCExport/Common/Protocols/LISA/SystemCCoprocBusProtocol.lisa`.

`SystemCCoprocBusProtocol` provides the following behaviors:

#### **accessIsNonSecure()**

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

#### **accessIsPriv()**

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

**addCoproprocessor()**

```
slave behavior addCoproprocessor(Coproprocessor*, int num) : void;
```

**removeCoproprocessor()**

```
slave behavior removeCoproprocessor(Coproprocessor*, int num) : void;
```

## 2.76 SystemCPChannel protocol

Defined in \$PVLIB\_HOME/examples/SystemCExport/Common/Protocols/LISA/SystemCPChannelProtocol.lisa.

### About SystemCPChannel protocol

Protocol used to communicate power state changes between a power controller and a device.

The `sg::PChannel::presp_t` enumeration provides two values:

- `sg::PChannel::ACCEPT`
- `sg::PChannel::DENY`

`SystemCPChannel` provides the following behaviors:

**pactive()**

```
master behavior pactive (uint32_t pstate) : void;
```

To be implemented by a power controller.

**prequest()**

```
slave behavior prequest (uint32_t pstate) : pchannel::presp_t;
```

To be implemented by a device.

## 2.77 SystemCoherencyInterface protocol

Defined in \$PVLIB\_HOME/LISA/SystemCoherencyInterface.lisa.

`SystemCoherencyInterface` provides the following behaviors:

**doDownstreamAction()**

```
optional slave behavior doDownstreamAction(const  
SystemCoherency::DownstreamAction&) : bool;
```

**doUpstreamAction()**

```
optional master behavior doUpstreamAction(const SystemCoherency::UpstreamAction&) :
    bool;
```

## 2.78 TZFilterControl protocol

Defined in \$PVLIB\_HOME/LISA/TZFilterUnit.lisa.

**About TZFilterControl protocol**

This protocol controls the communication between filter units and control registers in the APB control block.

TZFilterControl provides the following behaviors:

**checkPermission()**

```
optional slave behavior checkPermission(const pv::TransactionAttributes*
    attributes_,
                                       pv::bus_addr_t page_base_,
                                       bool is_read_,
                                       pv::RemapRequest& req_,
                                       bool & abort_on_error_) : bool;
```

Check the permission of the transactions filtered by the filter unit, using the information in the APB control block.

**isEnabled()**

```
slave behavior isEnabled() : bool;
```

Check if the filter unit is enabled or not. The APB control block controls the unit.

**isSecureSlave()**

```
optional slave behavior isSecureSlave() : bool;
```

Check if the connected slave is secure or not.

**setConfig()**

```
optional master behavior setConfig(bool rd_spec_enable, bool wr_spec_enable,
    uint32_t action) : void;
```

Pass the configurations to the filter.

## 2.79 TZSwitchControl protocol

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

### About TZSwitchControl protocol

Allow secure and normal TrustZone bus signals to be routed separately.

Transactions received on the TZSwitch `pvbus_input` slave port are routed according to a configuration that is set up using parameters and/or the control port. Separate rules can be given for secure and for normal transactions.

Transactions can be routed to one of the two master ports, `pvbus_port_a` or `pvbus_port_b`, can be ignored, or can generate aborts.

`TZSwitchControl` provides the following behaviors:

#### `routeAccesses ()`

```
slave behavior routeAccesses(TZSwitch_InputFilter input,
                             TZSwitch_RouteOption destination) : void;
```

This behavior takes two arguments:

- `input` selects which types of signals are reconfigured:
  - TZINPUT\_SECURE**  
Change the routing for secure transactions
  - TZINPUT\_NORMAL**  
Change the routing for normal transactions
  - TZINPUT\_ANY**  
Change the routing for all transactions
- `destination` selects how the chosen transactions are routed:
  - TZROUTE\_IGNORE**  
Transactions are ignored. Reads return 0.
  - TZROUTE\_TO\_PORT\_A**  
Route transactions to `pvbus_port_a`.
  - TZROUTE\_TO\_PORT\_B**  
Route transactions to `pvbus_port_b`.
  - TZROUTE\_ABORT**  
Cause transactions to generate an abort.

Initial routing is configured using TZSwitch parameters `secure` and `normal` based on the following values:

- 0**  
Ignore

- 1 Port A
- 2 Port B
- 3 Abort

Both default and explicit parameter values are overridden by any runtime calls to `routeAccesses()` on the control port.

## 2.80 TimerCallback protocol

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

### About TimerCallback protocol

When a `ClockTimer` reaches zero, it invokes the `signal()` behavior on its `timer_callback` port. This allows a component to process a timed callback by implementing this behavior on a slave port. The slave can also return a non-zero value to retrigger the timer.

`TimerCallback` provides the following behaviors:

#### **signal()**

```
peer behavior signal() : uint32_t;
```

Invoked when a clock timer reaches zero. If a non-zero value is returned, the clock restarts its countdown from the returned value.

## 2.81 TimerCallback64 protocol

Defined in `$PVLIB_HOME/LISA/TimerCallbackProtocol64.lisa`.

### About TimerCallback64 protocol

When a `ClockTimer` reaches zero, it invokes the `signal()` behavior on its `timer_callback` port. This allows a component to process a timed callback by implementing this behavior on a slave port. The slave can also return a non-zero value to retrigger the timer.

`TimerCallback64` provides the following behaviors:

#### **signal()**

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

Invoked when a clock timer reaches zero. If a non-zero value is returned, the clock restarts its countdown from the returned value.

## 2.82 TimerControl protocol

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

### About TimerControl protocol

This protocol controls the actions of the component. It permits a timer to be set to schedule a callback after a given number of ticks at the rate of the clock input.

If a timer is set while it is counting, it starts counting the new number of ticks without sending the original callback. Canceling a timer when it is not active has no effect.

TimerControl provides the following behaviors:

#### **cancel()**

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

Cancel the countdown on the active ClockTimer, preventing the callback from being invoked.

#### **isSet()**

```
slave behavior isSet() : bool;
```

Test whether the timer is currently actively counting.

#### **remaining()**

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

Return how many ticks remain before the timer's callback event will be signalled.

#### **set()**

```
slave behavior set(uint32_t ticks) : void;
```

Start the timer counting for the given number of ticks of its input clock. When the timer reaches zero, the scheduler invokes the `signal()` behaviour on its callback port, see [TimerCallback protocol](#).

## 2.83 TimerControl64 protocol

Defined in `$PVLIB_HOME/LISA/TimerControlProtocol64.lisa`.

### About TimerControl64 protocol

This protocol controls the actions of the component. It permits a timer to be set to schedule a callback after a given number of ticks at the rate of the clock input.

If a timer is set while it is counting, it starts counting the new number of ticks without sending the original callback. Canceling a timer when it is not active has no effect.

TimerControl64 provides the following behaviors:

#### **cancel()**

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

Cancel the countdown on the active ClockTimer64, preventing the callback from being invoked.

#### **isSet()**

```
slave behavior isSet() : bool;
```

Test whether the timer is currently actively counting.

#### **remaining()**

```
slave behavior remaining() : uint64_t;
```

Return how many ticks remain before the timer's callback event will be signalled.

#### **set()**

```
slave behavior set(uint64_t ticks) : void;
```

Start the timer counting for the given number of ticks of its input clock. When the timer reaches zero, the scheduler invokes the `signal` behaviour on its callback port, see [TimerCallback64 protocol](#).

## 2.84 VECBProtocol protocol

Defined in `$PVLIB_HOME/examples/LISA/Common/LISA/VECBProtocol.lisa`.

VECBProtocol provides the following behaviors:



**read()**

```
slave behavior read(const uint8_t function, const uint16_t device, uint32_t *  
data) : bool;
```

**write()**

```
slave behavior write(const uint8_t function, const uint16_t device, const uint32_t  
data) : bool;
```

## 2.85 VGICComponentTraceExport protocol

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

### About VGICComponentTraceExport protocol

This protocol is a workaround for a LISA problem where the CADI interface of the `VGIC_Component` is not exported, but we want to export the trace sources. The trace sources can be artificially exported onto another `sg::ComponentTrace` by using this interface and not exported to the `VGIC_Component` CADI interface.

To use the `VGIC_Component`'s `export_trace` port, you must set the parameter `export_trace_to_cadi` to false, otherwise the model aborts at run time.

`VGICComponentTraceExport` provides the following behaviors:

**exportTrace()**

```
optional slave behavior exportTrace(/*sg::ComponentTrace*/void*) : void;
```

## 2.86 VGICReportingProtocol protocol

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

`VGICReportingProtocol` provides the following behaviors:

**logErrors()**

```
optional slave behavior logErrors( const char* buffer_ ) : void;
```

**logFatal()**

```
optional slave behavior logFatal( const char* buffer_ ) : void;
```

**logWarnings()**

```
optional slave behavior logWarnings( const char* buffer_ ) : void;
```

**setEnables()**

```
optional master behavior setEnables( uint32_t new_enable_ ) : uint32_t;
```

Enable outputs on the behaviours above, returns the old value.

**bit[0]**

Log warnings enabled.

**bit[1]**

Log errors enabled.

**bit[2]**

Log fatal enabled.

## 2.87 Value protocol

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

**About Value protocol**

The Value protocol allows a master to send a 32-bit unsigned value to a slave.

`value` provides the following behaviors:

**setValue()**

```
optional peer behavior setValue(uint32_t /*value*/) : void;
```

Sets a 32-bit value for the signal.

## 2.88 ValueState protocol

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

**About ValueState protocol**

The ValueState protocol allows a master to retrieve the current value from a slave.

`valueState` provides the following behaviors:

**getValue()**

```
peer behavior getValue() : uint32_t;
```

Returns a 32-bit value for the signal.

**setValue()**

```
peer behavior setValue(uint32_t value) : void;
```

Sets a 32-bit value for the signal.

## 2.89 ValueState\_64 protocol

Defined in \$PVLIB\_HOME/LISA/Value64Protocol.lisa.

**About ValueState\_64 protocol**

The `valueState_64` protocol allows a master to retrieve the current value from a slave.

`valueState_64` provides the following behaviors:

**getValue()**

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

Returns a 64-bit value for the signal.

**setValue()**

```
peer behavior setValue(uint64_t value) : void;
```

Sets a 64-bit value for the signal.

## 2.90 Value\_64 protocol

Defined in \$PVLIB\_HOME/LISA/Value64Protocol.lisa.

**About Value\_64 protocol**

The `value_64` protocol allows a master to send a 64-bit unsigned value to a slave.

`value_64` provides the following behaviors:

**setValue()**

```
optional peer behavior setValue(uint64_t /*value*/) : void;
```

Sets a 64-bit value for the signal.

## 2.91 VirtualEthernet protocol

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

### About VirtualEthernet protocol

The Ethernet frame class encapsulates an Ethernet frame in a broken-up format that is more accessible by components. For information on the class definition, see the `EthernetFrame.h` header file located in `$PVLIB_HOME/include/components/VirtualEthernet/Protocol/`.

`VirtualEthernet` provides the following behaviors:

#### **sendToMaster()**

```
master behavior sendToMaster(EthernetFrame* frame) : void;
```

Send an Ethernet frame to the master port.

#### **sendToSlave()**

```
slave behavior sendToSlave(EthernetFrame* frame) : void;
```

Send an Ethernet frame to the slave port.

## 2.92 VisEventRecorderProtocol protocol

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

### About VisEventRecorderProtocol

The `VisEventRecorderProtocol` is used to play back and record events in the visualisation component of a platform system. The main purpose is for recording GUI benchmarks and regression tests for operating systems. A master port of this protocol is in the Visualisation component and a slave port is in the `VisEventRecorder` component.

`VisEventRecorderProtocol` provides the following behaviors:

#### **getEvent()**

```
slave behavior getEvent(VisEvent *event) : bool;
```

`processEvents()` is called to notify the master component, for example Visualisation, that new events are available.

The new events are retrieved by `getEvent()` from within `processEvents()`.

The slave component decides whether playback is enabled or disabled.

Playback events:

- Return true and fill `event` with the next event if there is one
- Return false if there is no event

It is safe to call this behavior from outside of `processEvents()`. If so, it always returns false.

### **processEvents()**

```
master behavior processEvents() : void;
```

The slave component calls this behavior in the master component to notify the master component that new events are now available and must be processed.

The new events should be retrieved using `getEvent()` from within `processEvents()`.

### **putEvent()**

```
slave behavior putEvent(const VisEvent *event) : void;
```

Record events call this behavior:

- Regardless of whether recording is enabled or disabled
- Even for events that just came from `getEvent()`

The slave component decides whether recording is enabled or disabled.

### **registerVisRegion()**

```
slave behavior registerVisRegion(VisRegion *region, const char *regionName) : void;
```

Called on initialisation. Associates names with `visRegion` pointers.

The slave component does not access the `visRegion` objects.

All `visRegion` objects, usually `visRenderRegion` and `visPushButtonRegion`, should be registered, but at least the ones where `visEvent::region` is used in the event loop.

Use the instance name for the region in the visualisation component as the name. For example `registerVisRegion(myRegion, "myRegion");`

## 2.93 v7\_VGIC\_Configuration\_Protocol protocol

Defined in `$PVLIB_HOME/LISA/v7_VGIC_Configuration_Protocol.lisa`.

`v7_VGIC_Configuration_Protocol` provides the following behaviors:

### **getNumberOfCores()**

```
slave behavior getNumberOfCores() : unsigned;
```

**setMasterIdToCoreNumberMapping()**

```
slave behavior setMasterIdToCoreNumberMapping(
    uint32_t master_id_,
    uint32_t master_id_mask_,
    unsigned cpu_interface_number_,
    unsigned inout_cluster_number_,
    unsigned inout_cpu_number_in_cluster_
) : bool;
```

## 2.94 v8EmbeddedCrossTrigger\_controlprotocol protocol

Defined in \$PVLIB\_HOME/LISA/v8EmbeddedCrossTrigger.lisa.

**About v8EmbeddedCrossTrigger\_controlprotocol protocol**

This protocol connects the Cross Trigger Interface (CTI) in processor components to platform-level Cross Trigger Matrix (CTM) components.

This opaque protocol is not exportable across a SystemC interface.

v8EmbeddedCrossTrigger\_controlprotocol provides the following behaviors:

**getComponentIdByte()**

```
master behavior getComponentIdByte(unsigned pidn) : uint8_t;
```

**getPeripheralIdByte()**

```
master behavior getPeripheralIdByte(unsigned pidn) : uint8_t;
```

**init()**

```
slave behavior init (unsigned number_of_triggers, unsigned intack_mask, unsigned
    number_of_claim_bits, bool has_software_lock, bool has_CTIDEVCTL) : void;
```

**initDelayedSysReg()**

```
optional slave behavior initDelayedSysReg(SynchronizeSysRegHelper*, bool, bool ) :
    void;
```

**initTrace()**

```
optional slave behavior initTrace(SystemRegUpdateTraceProbe*, bool*, int ) : void;
```

**isOSUnlockCatchEnabled()**

```
optional slave behavior isOSUnlockCatchEnabled() : bool;
```

**isResetCatchEnabled()**

```
optional slave behavior isResetCatchEnabled() : bool;
```

**reg\_read()**

```
slave behavior reg_read(bool is_memory_mapped, uint32_t addr, bool is_non_secure) :  
    uint32_t;
```

**reg\_write()**

```
slave behavior reg_write(bool is_memory_mapped, uint32_t addr, bool is_non_secure,  
    uint32_t data) : void;
```

**reset()**

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

**setValue\_inputTrigger()**

```
slave behavior setValue_inputTrigger(unsigned index, sg::Signal::State state) :  
    void;
```

**setValue\_outputTrigger()**

```
master behavior setValue_outputTrigger(unsigned index, sg::Signal::State state) :  
    void;
```

## 3. Fast Models components

This chapter describes all model components in Fast Models, organized by component type.

For each component, the documentation includes notes about using the model, describes any deviations in the model from the Technical Reference Manual (TRM), and describes the ports and parameters.

### 3.1 Component differences

This topic lists the new and changed components in this release.

#### Differences between 11.28.23 and 11.29.19

**Table 3-1: Components added**

Component	Quality level
<a href="#">3.5.33 ARM Cortex-A320CT</a> on page 1686	Preliminary support
<a href="#">3.10.12 CHBCR</a> on page 3687	Alpha support
<a href="#">3.7.9 CombinedMessagingUnitAE</a> on page 3444	N/A
<a href="#">3.6.2 FrameTracingComponent</a> on page 3378	Alpha support
<a href="#">3.10.56 MHU320AE</a> on page 4628	Full support
<a href="#">3.10.64 MMU_720AE</a> on page 4788	rOp0=pre
<a href="#">3.10.73 OTPW</a> on page 4907	Full support
<a href="#">3.7.60 TC25_SecureAccessConfig</a> on page 3557	N/A

**Table 3-2: Components removed**

Component
DP500
DP500x2
DP550
DP550x2
DP650
DP650x2
Mali_G51
Mali_G72
V550

**Table 3-3: Components changed**

Component	Has the IP revision changed?	Has the quality level changed?	Have ports been added or removed?	Have parameters been added or removed?
AEMvACT	No	No	No	Yes
ARM Cortex-A520AECT	No	No	No	Yes



Component	Has the IP revision changed?	Has the quality level changed?	Have ports been added or removed?	Have parameters been added or removed?
ARMCortexM55CT	No	No	Yes	Yes
ARMCortexR52CT	No	No	No	Yes
AddressTranslationUnit	No	No	No	Yes
CMN700	No	No	Yes	Yes
CMN_S3	No	No	Yes	Yes
D71	No	No	Yes	No
DebugROM	No	No	No	Yes
GIC700	No	No	No	Yes
GIC700_Filter	No	No	No	Yes
GIC720AE	No	No	No	Yes
GIC720AE_Filter	No	No	No	Yes
Labeller	No	No	No	Yes
LifeCycleManager	No	No	No	Yes
MMU_S3	No	No	No	Yes
Mali_G720	No	No	Yes	No
Mali_G725	No	No	Yes	Yes
MemoryMappedCounterModule	No	No	No	Yes
PASSwitch	No	No	No	Yes
PL370_HDLCD	No	No	Yes	No
PMU	No	No	No	Yes
SMMUv3AEM	No	No	No	Yes

## 3.2 Bridge components

This section describes the Bridge components.

These components allow conversion between the following protocols:

- PVBUS and AMBAPV.
- Signal and AMBAPVSignal.
- StateSignal and AMBAPVSignalState.
- Value(\_64) and AMBAPVValue(64).
- ValueState(\_64) and AMBAPVValueState(64).

LISA+ source for the bridge components is located in `$PVLIB_HOME/examples/SystemCEExport/Bridges/`.

The AMBAPV protocols and components are designed to interface with the AMBA® TLM PV library for ASI TLM 2.0. Fast Models provides this library as a standard way of mapping the AMBA protocol on top of ASI TLM 2.0.2 kit at PV level.

For more information about the AMBA TLM PV library for ASI TLM 2.0.2 kit, see the Fast Models documentation in `$MAXCORE_HOME/AMBA-PV/doc/`.

For more information about ASI TLM 2.0, see the Accellera documentation that is provided with the kit.

See also:

- [Accellera Systems Initiative](#)
- [AMBA-PV Extensions to TLM 2.0 Developer Guide](#)
- [Fast Models User Guide, SystemC Export with Multiple Instantiation](#)

### 3.2.1 AMBAPV2PVBUS

AMBA-PV to PVBUS protocol converter. This model is written in LISA+.

#### About AMBAPV2PVBUS

- PVBUS does not support transactions with `byte_enable` set (strobing transactions, in AXI terms). This bridge component rejects them.
- Variants of this component also exist with multiple input and output ports.

The bridge enables the ACP port to treat a transaction as coherent. It provides an additional parameter to specify the default shared bit value for incoming AMBA-PV transactions.

It also enables the shared bit to be specified by the “shareable” attribute of an AMBA-PV transaction using the `amba_pv_attributes` class. (Requires you to define the `AMBA_PV_INCLUDE_ATTRIBUTES` macro at compile time.)

#### Limitations

Fast Models bridges between PVBUS and AMBA-PV can transport Memory Tagging Extension (MTE) operations (tag stores, tag loads, and tag-checked loads and stores). These operations are transported opaquely, so the endpoint must be using PVBUS. This means you cannot handle these operations in your own TLM components.

#### Iris and MTI instances for AMBAPV2PVBUS

This model has the following Iris instances:

**Table 3-4: AMBAPV2PVBUS Iris instances**

InstanceName	ComponentName
AMBAPV2PVBUS	AMBAPV2PVBUS
AMBAPV2PVBUS.bus_master	PVBUSMaster

This model has the following MTI trace components:

**Table 3-5: AMBAPV2PVBUS MTI instances**

InstanceName	ComponentName
AMBAPV2PVBUS	AMBAPV2PVBUS
AMBAPV2PVBUS.bus_master	PVBUSMaster

### Ports for AMBAPV2PVBUS

**Table 3-6: Ports**

Name	Protocol	Type	Description
amba_pv_s	AMBAPV	Slave	-
pvbuss_m	PVBUS	Master	-

### Parameters for AMBAPV2PVBUS

#### **base\_addr**

##### Type

int

##### Default value

0x0

##### Description

Base address.

#### **report\_errors**

##### Type

bool

##### Default value

0x0

##### Description

Report transactions which do not comply with PVBUS protocol requirements.

#### **shareable**

##### Type

bool

##### Default value

0x1

##### Description

Shareable default.

### 3.2.2 AMBAPVACE2PVBUS

AMBA-PV ACE to PVBUS protocol converter. This model is written in LISA+.

#### About AMBAPVACE2PVBUS

- AMBAPVACE2PVBUS depends on the AMBA-PV API, which must be at least version 1.4.
- The translation of bus transactions by the bridge has some impact on performance. Bus masters that cache memory transactions avoid much of this impact. The bridge does not support DMI.
- PVBUS does not support transactions with `byte_enable` set (strobing transactions, in AXI terms). This bridge component rejects them.

#### Iris and MTI instances for AMBAPVACE2PVBUS

This model has the following Iris instances:

**Table 3-7: AMBAPVACE2PVBUS Iris instances**

InstanceName	ComponentName
AMBAPVACE2PVBUS	AMBAPVACE2PVBUS
AMBAPVACE2PVBUS.bus_master	PVBUSMaster

This model has the following MTI trace components:

**Table 3-8: AMBAPVACE2PVBUS MTI instances**

InstanceName	ComponentName
AMBAPVACE2PVBUS	AMBAPVACE2PVBUS
AMBAPVACE2PVBUS.bus_master	PVBUSMaster

#### Ports for AMBAPVACE2PVBUS

**Table 3-9: Ports**

Name	Protocol	Type	Description
amba_pv_ace_s	AMBAPVACE	Slave	-
pvbust_m	PVBUS	Master	-

#### Parameters for AMBAPVACE2PVBUS

##### **report\_errors**

##### Type

bool

##### Default value

0x0

##### Description

Report transactions which do not comply with PVBUS protocol requirements.

### 3.2.3 AMBAPVSignal2SGSignal

AMBA-PV Signal to SystemGenerator Signal protocol converter. This model is written in LISA+.



Note

Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for AMBAPVSignal2SGSignal

This model has the following Iris instances:

**Table 3-10: AMBAPVSignal2SGSignal Iris instances**

InstanceName	ComponentName
AMBAPVSignal2SGSignal	AMBAPVSignal2SGSignal

#### Ports for AMBAPVSignal2SGSignal

**Table 3-11: Ports**

Name	Protocol	Type	Description
amba_pv_signal_s	AMBAPVSignal	Slave	Input slave port for connection from top-level AMBAPVSignal slave port.
sg_signal_m	Signal	Master	Handles outgoing signal state changes. Converted signal state changes are sent out through this port.

### 3.2.4 AMBAPVSignalState2SGStateSignal

AMBA-PV SignalState to SystemGenerator StateSignal protocol converter. This model is written in LISA+.



Note

Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for AMBAPVSignalState2SGStateSignal

This model has the following Iris instances:

**Table 3-12: AMBAPVSignalState2SGStateSignal Iris instances**

InstanceName	ComponentName
AMBAPVSignalState2SGStateSignal	AMBAPVSignalState2SGStateSignal

## Ports for AMBAPVSignalState2SGStateSignal

**Table 3-13: Ports**

Name	Protocol	Type	Description
amba_pv_signal_s	AMBAPVSignalState	Slave	-
sg_signal_m	StateSignal	Master	-

### 3.2.5 AMBAPVValue2SGValue

AMBA-PV Value to SystemGenerator Value protocol converter. This model is written in LISA+.



Note

Variants of this component also exist with multiple input and output ports.

## Iris and MTI instances for AMBAPVValue2SGValue

This model has the following Iris instances:

**Table 3-14: AMBAPVValue2SGValue Iris instances**

InstanceName	ComponentName
AMBAPVValue2SGValue	AMBAPVValue2SGValue

## Ports for AMBAPVValue2SGValue

**Table 3-15: Ports**

Name	Protocol	Type	Description
amba_pv_value_s	AMBAPVValue	Slave	-
sg_value_m	Value	Master	-

### 3.2.6 AMBAPVValue2SGValue64

AMBA-PV Value64 to SystemGenerator Value\_64 protocol converter. This model is written in LISA+.



Note

Variants of this component also exist with multiple input and output ports.

## Iris and MTI instances for AMBAPVValue2SGValue64

This model has the following Iris instances:

**Table 3-16: AMBAPVValue2SGValue64 Iris instances**

InstanceName	ComponentName
AMBAPVValue2SGValue64	AMBAPVValue2SGValue64

### Ports for AMBAPVValue2SGValue64

**Table 3-17: Ports**

Name	Protocol	Type	Description
amba_pv_value_s	AMBAPVValue64	Slave	-
sg_value_m	Value_64	Master	-

## 3.2.7 AMBAPVValue642SMMUv3AEMIdentify

AMBA-PV Value64 to SMMUv3AEMIdentify protocol converter. This model is written in LISA+.

### Iris and MTI instances for AMBAPVValue642SMMUv3AEMIdentify

This model has the following Iris instances:

**Table 3-18: AMBAPVValue642SMMUv3AEMIdentify Iris instances**

InstanceName	ComponentName
AMBAPVValue642SMMUv3AEMIdentify	AMBAPVValue642SMMUv3AEMIdentify

### Ports for AMBAPVValue642SMMUv3AEMIdentify

**Table 3-19: Ports**

Name	Protocol	Type	Description
identify	SMMUv3AEMIdentifyProtocol	Master	-
identify_reply	AMBAPVValue64	Master	-
identify_request	AMBAPVValue64	Slave	-

## 3.2.8 AMBAPVValue642VECB

AMBA-PV to VECB protocol converter. This model is written in LISA+.

### Iris and MTI instances for AMBAPVValue642VECB

This model has the following Iris instances:

**Table 3-20: AMBAPVValue642VECB Iris instances**

InstanceName	ComponentName
AMBAPVValue642VECB	AMBAPVValue642VECB

## Ports for AMBAPVValue642VECB

**Table 3-21: Ports**

Name	Protocol	Type	Description
amba_pv_ctrl_s	AMBAPVValue	Slave	-
amba_pv_data_s	AMBAPVValue64	Slave	-
vecb_m	VECBProtocol	Master	-

### 3.2.9 AMBAPVValueState2SGValueState

AMBA-PV ValueState to SystemGenerator ValueState protocol converter. This model is written in LISA+.



Note

Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for AMBAPVValueState2SGValueState

This model has the following Iris instances:

**Table 3-22: AMBAPVValueState2SGValueState Iris instances**

InstanceName	ComponentName
AMBAPVValueState2SGValueState	AMBAPVValueState2SGValueState

## Ports for AMBAPVValueState2SGValueState

**Table 3-23: Ports**

Name	Protocol	Type	Description
amba_pv_value_s	AMBAPVValueState	Slave	-
sg_value_m	ValueState	Master	-

### 3.2.10 AMBAPVValueState2SGValueState64

AMBA-PV ValueState64 to SystemGenerator ValueState\_64 protocol converter. This model is written in LISA+.



Note

Variants of this component also exist with multiple input and output ports.



## Iris and MTI instances for AMBAPVValueState2SGValueState64

This model has the following Iris instances:

**Table 3-24: AMBAPVValueState2SGValueState64 Iris instances**

InstanceName	ComponentName
AMBAPVValueState2SGValueState64	AMBAPVValueState2SGValueState64

## Ports for AMBAPVValueState2SGValueState64

**Table 3-25: Ports**

Name	Protocol	Type	Description
amba_pv_value_s	AMBAPVValueState64	Slave	-
sg_value_m	ValueState_64	Master	-

## 3.2.11 BroadcastSignal2AMBAPVSignal

Broadcast signal to AMBAPVSignal converter. This model is written in LISA+.

## Iris and MTI instances for BroadcastSignal2AMBAPVSignal

This model has the following Iris instances:

**Table 3-26: BroadcastSignal2AMBAPVSignal Iris instances**

InstanceName	ComponentName
BroadcastSignal2AMBAPVSignal	BroadcastSignal2AMBAPVSignal

## Ports for BroadcastSignal2AMBAPVSignal

**Table 3-27: Ports**

Name	Protocol	Type	Description
amba_pv_signal_m	AMBAPVSignal	Master	-
amba_pv_signal_s	AMBAPVSignal	Slave	-
b_signal	Signal	Broadcast	-

## 3.2.12 Clock2SystemC

Clock to SystemC Converter. This model is written in LISA+.

## Iris and MTI instances for Clock2SystemC

This model has the following Iris instances:

**Table 3-28: Clock2SystemC Iris instances**

InstanceName	ComponentName
Clock2SystemC	Clock2SystemC

## Ports for Clock2SystemC

**Table 3-29: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	-
current_ticks_s	<a href="#">AMBAPVValueState64</a>	Slave	-
get_clock_s	<a href="#">AMBAPVValueState64</a>	Slave	-
rate_in_Hz_s	<a href="#">AMBAPVValueState64</a>	Slave	-
set_clock_m	<a href="#">AMBAPVValue64</a>	Master	-

## 3.2.13 ClockRateConversion

ClockRateControl to rate in Hz (Value\_64) Converter. This model is written in LISA+.

### Iris and MTI instances for ClockRateConversion

This model has the following Iris instances:

**Table 3-30: ClockRateConversion Iris instances**

InstanceName	ComponentName
ClockRateConversion	ClockRateConversion
ClockRateConversion.clk_div0	ClockDivider
ClockRateConversion.clk_div1	ClockDivider
ClockRateConversion.clk_div2	ClockDivider
ClockRateConversion.clk_div3	ClockDivider

This model has the following MTI trace components:

**Table 3-31: ClockRateConversion MTI instances**

InstanceName	ComponentName
ClockRateConversion.clk_div0	<a href="#">ClockDivider</a>
ClockRateConversion.clk_div1	<a href="#">ClockDivider</a>
ClockRateConversion.clk_div2	<a href="#">ClockDivider</a>
ClockRateConversion.clk_div3	<a href="#">ClockDivider</a>

## Ports for ClockRateConversion

**Table 3-32: Ports**

Name	Protocol	Type	Description
clock	<a href="#">ClockSignal</a>	Slave	-
rate_ctrl[4]	<a href="#">ClockRateControl</a>	Slave	-
rate_hz[4]	<a href="#">Value_64</a>	Master	-

## Parameters for ClockRateConversion

### `clk_div0.div`

**Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

### `clk_div0.mul`

**Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

### `clk_div1.div`

**Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

### `clk_div1.mul`

**Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

### `clk_div2.div`

**Type**

int

**Default value**

0x1

**Description**  
Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clk\_div2.mul**

**Type**  
int

**Default value**  
0x1

**Description**  
Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clk\_div3.div**

**Type**  
int

**Default value**  
0x1

**Description**  
Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clk\_div3.mul**

**Type**  
int

**Default value**  
0x1

**Description**  
Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

3.2.14 ClockSignal2SC\_ClockSignal

ClockSignal to SystemC ClockSignal converter. This model is written in LISA+.

Iris and MTI instances for ClockSignal2SC\_ClockSignal

This model has the following Iris instances:

Table 3-33: ClockSignal2SC\_ClockSignal Iris instances

InstanceName	ComponentName
ClockSignal2SC_ClockSignal	ClockSignal2SC_ClockSignal

## Ports for ClockSignal2SC\_ClockSignal

**Table 3-34: Ports**

Name	Protocol	Type	Description
clk_out	ClockSignal	Slave	-
sc_clk_out	SC_ClockSignal	Master	-

## 3.2.15 CoprocBus2SystemC

CoprocBusProtocol to SystemCCoprocBusProtocol converter. This model is written in LISA+.

### Iris and MTI instances for CoprocBus2SystemC

This model has the following Iris instances:

**Table 3-35: CoprocBus2SystemC Iris instances**

InstanceName	ComponentName
CoprocBus2SystemC	CoprocBus2SystemC

## Ports for CoprocBus2SystemC

**Table 3-36: Ports**

Name	Protocol	Type	Description
coproc_bus_s	CoprocBusProtocol	Slave	-
sc_coproc_bus_m	SystemCCoprocBusProtocol	Master	-

## 3.2.16 CounterInterface2SystemC

CounterInterface to SystemC Converter. This model is written in LISA+.

### Iris and MTI instances for CounterInterface2SystemC

This model has the following Iris instances:

**Table 3-37: CounterInterface2SystemC Iris instances**

InstanceName	ComponentName
CounterInterface2SystemC	CounterInterface2SystemC

## Ports for CounterInterface2SystemC

**Table 3-38: Ports**

Name	Protocol	Type	Description
amba_pv_eventUpdate_m	AMBAPVValue	Master	-
amba_pv_getCounterValue_s	AMBAPVValueState64	Slave	-
amba_pv_requestEventUpdate_s	AMBAPVValue64	Slave	-
amba_pv_requestSignalUpdate_s	AMBAPVValue64	Slave	-

Name	Protocol	Type	Description
amba_pv_setEnabled_m	AMBAPVValue	Master	-
amba_pv_signalUpdate_m	AMBAPVValue	Master	-
cntvalueb	CounterInterface	Slave	-

### 3.2.17 InstructionCount2SystemC

InstructionCount to SystemC Converter. This model is written in LISA+.



Note

Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for InstructionCount2SystemC

This model has the following Iris instances:

**Table 3-39: InstructionCount2SystemC Iris instances**

InstanceName	ComponentName
InstructionCount2SystemC	InstructionCount2SystemC

#### Ports for InstructionCount2SystemC

**Table 3-40: Ports**

Name	Protocol	Type	Description
inst_count	AMBAPVValueState64	Slave	-
run_state	AMBAPVValueState	Slave	-
ticks	InstructionCount	Slave	-

### 3.2.18 LCD2SystemC

Converts LCD protocol to SystemC. This model is written in LISA+.

#### Iris and MTI instances for LCD2SystemC

This model has the following Iris instances:

**Table 3-41: LCD2SystemC Iris instances**

InstanceName	ComponentName
LCD2SystemC	LCD2SystemC

## Ports for LCD2SystemC

**Table 3-42: Ports**

Name	Protocol	Type	Description
all_received_sPL	AMBAPVSignal	Master	-
all_received_u	AMBAPVSignal	Master	-
lcd_s	LCD	Slave	-
lock_m	AMBAPVValueState64	Master	-
setPreferredLayout_d	AMBAPVValue	Master	-
setPreferredLayout_h	AMBAPVValue	Master	-
setPreferredLayout_w	AMBAPVValue	Master	-
unlock_m	AMBAPVSignal	Master	-
update_h	AMBAPVValue	Master	-
update_w	AMBAPVValue	Master	-
update_x	AMBAPVValue	Master	-
update_y	AMBAPVValue	Master	-

## 3.2.19 PChannel2SystemC

PChannel to SystemC Converter. This model is written in LISA+.

### Iris and MTI instances for PChannel2SystemC

This model has the following Iris instances:

**Table 3-43: PChannel2SystemC Iris instances**

InstanceName	ComponentName
PChannel2SystemC	PChannel2SystemC

## Ports for PChannel2SystemC

**Table 3-44: Ports**

Name	Protocol	Type	Description
pchannel	PChannel	Slave	-
sc_pchannel	SystemCPChannel	Master	-

### 3.2.20 PVBUS2AMBAPV

PVBus to AMBA-PV protocol converter. This model is written in LISA+.

#### About PVBUS2AMBAPV



Variants of PVBUS2AMBAPV also exist with multiple input and output ports.

The AMBAPV protocol definition in LISA, `AMBAPVProtocol.lisa`, specifies a 64-bit bus width, so the PVBUS2AMBAPV bridge also handles a 64-bit bus width.

If you need to connect to a component that uses a bus interface with a smaller or larger bus width, the recommended method is to insert a downsizer or upsizer respectively.

Alternatively, you could define a new bus protocol with the required bit width, for example AMBAPV32, and update the corresponding bridges to use the new protocol on AMBA-PV ports:

```
master port<AMBAPV32> amba_pv_m
```

#### Limitations

Fast Models bridges between PVBus and AMBA-PV can transport Memory Tagging Extension (MTE) operations (tag stores, tag loads, and tag-checked loads and stores).

These operations are transported opaquely, so the endpoint must be using PVBus. This means you cannot handle these operations in your own TLM components.

#### Dumping the DMI cache

DMI viewer provides the debugging functionality of the PVBUS2AMBAPV bridge. When activated, it dumps the content of the DMI cache in the bridge in the following CSV format:

```
Range_start, Range_end_incl, Pointer, Latency, R/W, Attributes
```

To activate this functionality, a name for the counters output file must be set, using the `counters-file-name` parameter. If the counters file name is set, when `dump-dmi-cache` is set to 1 at runtime, the DMI cache of the bridge is dumped. The runtime parameter is always reset to 0 when the dump has completed.

#### Iris and MTI instances for PVBUS2AMBAPV

This model has the following Iris instances:

**Table 3-45: PVBUS2AMBAPV Iris instances**

InstanceName	ComponentName
PVBUS2AMBAPV	PVBUS2AMBAPV



InstanceName	ComponentName
PVBus2AMBAPV.bus_bridge	PVBusBridge

This model has the following MTI trace components:

Table 3-46: PVBus2AMBAPV MTI instances

InstanceName	ComponentName
PVBus2AMBAPV	PVBus2AMBAPV
PVBus2AMBAPV.bus_bridge	PVBusBridge

Ports for PVBus2AMBAPV

Table 3-47: Ports

Name	Protocol	Type	Description
amba_pv_m	AMBAPV	Master	-
pdbus_s	PVBus	Slave	-

Parameters for PVBus2AMBAPV

counters-file-name

Type

string

Default value

""

Description

Prefix of the file name to store counters at the end of simulation.

dump-dmi-cache

Type

bool

Default value

0x0

Description

Dumps the content of the DMI cache into a file.

dump-dmi-file-name

Type

string

Default value

""

Description

Prefix of the file name to dump the content of the DMI when requested.

**force-dmi-size****Type**

bool

**Default value**

0x1

**Description**

Force DMI start and end address to be 4kB-aligned.

**min-range-to-cache****Type**

int

**Default value**

0x10000

**Description**

Min DMI range size to cache in the bridge.

**size****Type**

int

**Default value**

0x100000000000000

**Description**

Maximum size of memory region.

### 3.2.21 PVBUS2AMBAPVACE

PVBus to AMBA-PV ACE protocol converter. This model is written in LISA+.

#### About PVBUS2AMBAPVACE

PVBUS2AMBAPVACE depends on the AMBA-PV API, which must be at least version 1.4.

The translation of bus transactions by the bridge has some impact on performance. Bus masters that cache memory transactions avoid much of this impact.

DMI viewer provides the debugging functionality of the PVBUS2AMBAPV bridge. When activated, it dumps the content of the DMI cache in the bridge in the following CSV format:

```
Range start, Range end incl, Pointer, Latency, R/W, Attributes
```

To activate this functionality, set a name for the counters output file using the `counters-file-name` parameter. If the counters file name is set, when `dump-dmi-cache` is set to 1 at runtime, the DMI

cache of the bridge is dumped. The runtime parameter is always reset to 0 when the dump has completed.

## Iris and MTI instances for PVBUS2AMBAPVACE

This model has the following Iris instances:

**Table 3-48: PVBUS2AMBAPVACE Iris instances**

InstanceName	ComponentName
PVBUS2AMBAPVACE	PVBUS2AMBAPVACE
PVBUS2AMBAPVACE.bus_bridge	PVBUSBridge
PVBUS2AMBAPVACE.pvbus_tlm_switch	PVBUSMapper

This model has the following MTI trace components:

**Table 3-49: PVBUS2AMBAPVACE MTI instances**

InstanceName	ComponentName
PVBUS2AMBAPVACE	PVBUS2AMBAPVACE
PVBUS2AMBAPVACE.bus_bridge	PVBUSBridge
PVBUS2AMBAPVACE.pvbus_tlm_switch	PVBUSMapper

## Ports for PVBUS2AMBAPVACE

**Table 3-50: Ports**

Name	Protocol	Type	Description
amba_pv_ace_m	AMBAPVACE	Master	-
pvbus_over_tlm_control	PVBUSOverTLMControl	Slave	-
pvbus_s	PVBUS	Slave	-

## Parameters for PVBUS2AMBAPVACE

### counters-file-name

#### Type

string

#### Default value

""

#### Description

Prefix of the file name to store counters at the end of simulation.

### dmi-cache-name

#### Type

string

#### Default value

""

**Description**

DEPRECATED: This parameter will be ignored. Name of the DMI cache. Useful for multiple bridges to share the same cache.

**dump-dmi-cache****Type**

bool

**Default value**

0x0

**Description**

Dumps the content of the DMI cache into a file.

**dump-dmi-file-name****Type**

string

**Default value**

""

**Description**

Prefix of the file name to dump the content of the DMI when requested.

**force-dmi-size****Type**

bool

**Default value**

0x1

**Description**

Force DMI start and end address to be 4kB-aligned.

**min-range-to-cache****Type**

int

**Default value**

0x10000

**Description**

Min DMI range size to cache in the bridge.

**route-tlm****Type**

bool

**Default value**

0x1

**Description**

Route all the PVBUS traffic explicitly to the TLM bus. Allows to monitor transactions on the TLM bus but slows down the emulation. The routing must always be to TLM if there is not a corresponding AMBAPVACE2PVBUS bridge downstream.

**route-tlm-filter****Type**

string

**Default value**

""

Route TLM filter set a range (or multiple ranges) of addresses that will use PVBUS even if `route-tlm` is set to true.

The `route-tlm-filter` is specified in JSON format. Example,

```
[
  {
    "begin":0x2f000000',
    "size":0x1000
  },
  {
    "begin":0x4f000000',
    "size":0x2000
  }
]
```

**set-ace-lite****Type**

bool

**Default value**

0x0

**Description**

Set bridge mode when connecting to ace-lite ports. If true, the bridge will not deal with SNOOPS.

**size****Type**

int

**Default value**

0x10000000000000

**Description**

Maximum size of memory region, i.e. the first unsupported address.

### 3.2.22 PVBusBridge

A PVBusBridge bridges incoming transactions to a PVDevice port. This model is written in C++.

#### Iris and MTI instances for PVBusBridge

This model has the following Iris instances:

**Table 3-51: PVBusBridge Iris instances**

InstanceName	ComponentName
PVBusBridge	PVBusBridge

This model has the following MTI trace components:

**Table 3-52: PVBusBridge MTI instances**

InstanceName	ComponentName
PVBusBridge	PVBusBridge

#### Ports for PVBusBridge

**Table 3-53: Ports**

Name	Protocol	Type	Description
control	PVBusBridgeControl	Slave	Control signal.
device	PVDevice	Master	Optimised connection out to devices.
dump_dmi	Signal	Slave	On the assert of this signal the bridge will dump dmi cache content into a csv file
pvbus_s	PVBus	Slave	Connection in from bus master.
reset	Signal	Slave	On the assert of this signal, a reset of the bus slave will be latched this is used by the bus deadlock detection logic.

### 3.2.23 SC\_ClockSignal2ClockSignal

SystemC ClockSignal to ClockSignal converter. This model is written in LISA+.

#### Iris and MTI instances for SC\_ClockSignal2ClockSignal

This model has the following Iris instances:

**Table 3-54: SC\_ClockSignal2ClockSignal Iris instances**

InstanceName	ComponentName
SC_ClockSignal2ClockSignal	SC_ClockSignal2ClockSignal

#### Ports for SC\_ClockSignal2ClockSignal

**Table 3-55: Ports**

Name	Protocol	Type	Description
clk_in	ClockSignal	Master	-

Name	Protocol	Type	Description
sc_clk_in	SC_ClockSignal	Slave	-

### 3.2.24 SGSignal2AMBAPVSignal

SystemGenerator Signal to AMBA-PV Signal protocol converter. This model is written in LISA+.

#### About SGSignal2AMBAPVSignal

SGSignal to AMBA-PV signal protocol converter.



Note

Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for SGSignal2AMBAPVSignal

This model has the following Iris instances:

**Table 3-56: SGSignal2AMBAPVSignal Iris instances**

InstanceName	ComponentName
SGSignal2AMBAPVSignal	SGSignal2AMBAPVSignal

#### Ports for SGSignal2AMBAPVSignal

**Table 3-57: Ports**

Name	Protocol	Type	Description
amba_pv_signal_m	AMBAPVSignal	Master	Output master port for connection to top-level AMBAPVSignal master port. Converted signal state changes are sent out through this port.
sg_signal_s	Signal	Slave	Handles incoming signal state changes.

### 3.2.25 SGStateSignal2AMBAPVSignalState

SystemGenerator StateSignal to AMBA-PV SignalState protocol converter. This model is written in LISA+.

#### About SGStateSignal2AMBAPVSignalState

SGStateSignal to AMBA-PV SignalState protocol converter.



Note

Variants of this component also exist with multiple input and output ports.

## Iris and MTI instances for SGStateSignal2AMBAPVSignalState

This model has the following Iris instances:

**Table 3-58: SGStateSignal2AMBAPVSignalState Iris instances**

InstanceName	ComponentName
SGStateSignal2AMBAPVSignalState	SGStateSignal2AMBAPVSignalState

## Ports for SGStateSignal2AMBAPVSignalState

**Table 3-59: Ports**

Name	Protocol	Type	Description
amba_pv_signal_m	AMBAPVSignalState	Master	Output master port for connection to top-level AMBAPVValue64 master port. Converted value changes are sent out through this port.
sg_signal_s	StateSignal	Slave	Handles incoming value changes.

## 3.2.26 SGValue2AMBAPVValue

SystemGenerator Value to AMBA-PV Value protocol converter. This model is written in LISA+.

### About SGValue2AMBAPVValue



Variants of this component also exist with multiple input and output ports.

## Iris and MTI instances for SGValue2AMBAPVValue

This model has the following Iris instances:

**Table 3-60: SGValue2AMBAPVValue Iris instances**

InstanceName	ComponentName
SGValue2AMBAPVValue	SGValue2AMBAPVValue

## Ports for SGValue2AMBAPVValue

**Table 3-61: Ports**

Name	Protocol	Type	Description
amba_pv_value_m	AMBAPVValue	Master	Output master port for connection to top-level AMBAPVValue master port. Converted value changes are sent out through this port.
sg_value_s	Value	Slave	Handles incoming value changes.



### 3.2.27 SGValue2AMBAPVValue64

SystemGenerator Value\_64 to AMBA-PV Value64 protocol converter. This model is written in LISA+.

#### About SGValue2AMBAPVValue64



Note

Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for SGValue2AMBAPVValue64

This model has the following Iris instances:

**Table 3-62: SGValue2AMBAPVValue64 Iris instances**

InstanceName	ComponentName
SGValue2AMBAPVValue64	SGValue2AMBAPVValue64

#### Ports for SGValue2AMBAPVValue64

**Table 3-63: Ports**

Name	Protocol	Type	Description
amba_pv_value_m	AMBAPVValue64	Master	Output master port for connection to top-level AMBAPVValue master port. Converted value changes are sent out through this port.
sg_value_s	Value_64	Slave	Handles incoming value changes.

### 3.2.28 SGValueState2AMBAPVValueState

SystemGenerator ValueState to AMBA-PV ValueState protocol converter. This model is written in LISA+.

#### About SGValueState2AMBAPVValueState



Note

Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for SGValueState2AMBAPVValueState

This model has the following Iris instances:

**Table 3-64: SGValueState2AMBAPVValueState Iris instances**

InstanceName	ComponentName
SGValueState2AMBAPVValueState	SGValueState2AMBAPVValueState

### Ports for SGValueState2AMBAPVValueState

**Table 3-65: Ports**

Name	Protocol	Type	Description
amba_pv_value_m	AMBAPVValueState	Master	Output master port for connection to top-level AMBAPVValue master port. Converted value changes are sent out through this port.
sg_value_s	ValueState	Slave	Handles incoming value changes.

## 3.2.29 SGValueState2AMBAPVValueState64

SystemGenerator ValueState\_64 to AMBA-PV ValueState64 protocol converter. This model is written in LISA+.

### About SGValueState2AMBAPVValueState64



Note

Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGValueState2AMBAPVValueState64

This model has the following Iris instances:

**Table 3-66: SGValueState2AMBAPVValueState64 Iris instances**

InstanceName	ComponentName
SGValueState2AMBAPVValueState64	SGValueState2AMBAPVValueState64

### Ports for SGValueState2AMBAPVValueState64

**Table 3-67: Ports**

Name	Protocol	Type	Description
amba_pv_value_m	AMBAPVValueState64	Master	Output master port for connection to top-level AMBAPVValue64 master port. Converted value changes are sent out through this port.
sg_value_s	ValueState_64	Slave	Handles incoming value changes.

### 3.2.30 SMMUv3AEMIdentify2AMBAPVValue64

SMMUv3AEMIdentify to AMBA-PV Value64 protocol converter. This model is written in LISA+.

#### Iris and MTI instances for SMMUv3AEMIdentify2AMBAPVValue64

This model has the following Iris instances:

**Table 3-68: SMMUv3AEMIdentify2AMBAPVValue64 Iris instances**

InstanceName	ComponentName
SMMUv3AEMIdentify2AMBAPVValue64	SMMUv3AEMIdentify2AMBAPVValue64

#### Ports for SMMUv3AEMIdentify2AMBAPVValue64

**Table 3-69: Ports**

Name	Protocol	Type	Description
identify	<a href="#">SMMUv3AEMIdentifyProtocol</a>	Slave	SMMUv3AEMIdentifyProtocol input.
identify_reply	<a href="#">AMBAPVValue64</a>	Slave	From SystemC.
identify_request	<a href="#">AMBAPVValue64</a>	Master	To SystemC.

### 3.2.31 SystemC2Clock

Clock to SystemC Converter. This model is written in LISA+.

#### Iris and MTI instances for SystemC2Clock

This model has the following Iris instances:

**Table 3-70: SystemC2Clock Iris instances**

InstanceName	ComponentName
SystemC2Clock	SystemC2Clock

#### Ports for SystemC2Clock

**Table 3-71: Ports**

Name	Protocol	Type	Description
clk_out	<a href="#">ClockSignal</a>	Master	ClockSignal ouput
current_ticks_m	<a href="#">AMBAPVValueState64</a>	Master	To SystemC.
get_clock_m	<a href="#">AMBAPVValueState64</a>	Master	To SystemC.
rate_in_Hz_m	<a href="#">AMBAPVValueState64</a>	Master	To SystemC.
set_clock_s	<a href="#">AMBAPVValue64</a>	Slave	From SystemC.

### 3.2.32 SystemC2CprocBus

SystemCCoprocBusProtocol to CoprocBusProtocol converter. This model is written in LISA+.

#### Iris and MTI instances for SystemC2CprocBus

This model has the following Iris instances:

**Table 3-72: SystemC2CprocBus Iris instances**

InstanceName	ComponentName
SystemC2CprocBus	SystemC2CprocBus

#### Ports for SystemC2CprocBus

**Table 3-73: Ports**

Name	Protocol	Type	Description
coproc_bus_m	CoprocBusProtocol	Master	-
sc_coproc_bus_s	SystemCCoprocBusProtocol	Slave	-

### 3.2.33 SystemC2CounterInterface

SystemC to CounterInterface Converter. This model is written in LISA+.

#### Iris and MTI instances for SystemC2CounterInterface

This model has the following Iris instances:

**Table 3-74: SystemC2CounterInterface Iris instances**

InstanceName	ComponentName
SystemC2CounterInterface	SystemC2CounterInterface

#### Ports for SystemC2CounterInterface

**Table 3-75: Ports**

Name	Protocol	Type	Description
amba_pv_eventUpdate_s	AMBAPVValue	Slave	From SystemC.
amba_pv_getCounterValue_m	AMBAPVValueState64	Master	To SystemC.
amba_pv_requestEventUpdate_m	AMBAPVValue64	Master	To SystemC.
amba_pv_requestSignalUpdate_m	AMBAPVValue64	Master	To SystemC.
amba_pv_setEnabled_s	AMBAPVValue	Slave	From SystemC.
amba_pv_signalUpdate_s	AMBAPVValue	Slave	From SystemC.
cntvalueb	CounterInterface	Master	-

### 3.2.34 SystemC2InstructionCount

SystemC to InstructionCount Converter. This model is written in LISA+.



Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for SystemC2InstructionCount

This model has the following Iris instances:

**Table 3-76: SystemC2InstructionCount Iris instances**

InstanceName	ComponentName
SystemC2InstructionCount	SystemC2InstructionCount

#### Ports for SystemC2InstructionCount

**Table 3-77: Ports**

Name	Protocol	Type	Description
inst_count	AMBAPVValueState64	Master	To SystemC to request instruction count.
run_state	AMBAPVValueState	Master	To SystemC to request run state.
ticks	InstructionCount	Master	InstructionCount input.

### 3.2.35 SystemC2LCD

Converts SystemC to LCD protocol. This model is written in LISA+.

#### Iris and MTI instances for SystemC2LCD

This model has the following Iris instances:

**Table 3-78: SystemC2LCD Iris instances**

InstanceName	ComponentName
SystemC2LCD	SystemC2LCD

#### Ports for SystemC2LCD

**Table 3-79: Ports**

Name	Protocol	Type	Description
all_received_spl	AMBAPVSignal	Slave	From SystemC.
all_received_u	AMBAPVSignal	Slave	From SystemC.
lcd_m	LCD	Master	LCD output.
lock_s	AMBAPVValueState64	Slave	From SystemC.
setPreferredLayout_d	AMBAPVValue	Slave	From SystemC.

Name	Protocol	Type	Description
setPreferredLayout_h	AMBAPVValue	Slave	From SystemC.
setPreferredLayout_w	AMBAPVValue	Slave	From SystemC.
unlock_s	AMBAPVSignal	Slave	From SystemC.
update_h	AMBAPVValue	Slave	From SystemC.
update_w	AMBAPVValue	Slave	From SystemC.
update_x	AMBAPVValue	Slave	From SystemC.
update_y	AMBAPVValue	Slave	From SystemC.

### 3.2.36 SystemC2PChannel

SystemC to PChannel Converter. This model is written in LISA+.

#### Iris and MTI instances for SystemC2PChannel

This model has the following Iris instances:

**Table 3-80: SystemC2PChannel Iris instances**

InstanceName	ComponentName
SystemC2PChannel	SystemC2PChannel

#### Ports for SystemC2PChannel

**Table 3-81: Ports**

Name	Protocol	Type	Description
pchannel	PChannel	Master	-
sc_pchannel	SystemCPChannel	Slave	-

### 3.2.37 SystemC2VirtualEthernet

SystemC to VirtualEthernet Converter. This model is written in LISA+.

#### Iris and MTI instances for SystemC2VirtualEthernet

This model has the following Iris instances:

**Table 3-82: SystemC2VirtualEthernet Iris instances**

InstanceName	ComponentName
SystemC2VirtualEthernet	SystemC2VirtualEthernet

#### Ports for SystemC2VirtualEthernet

**Table 3-83: Ports**

Name	Protocol	Type	Description
virtualethernet_m	VirtualEthernet	Master	-

Name	Protocol	Type	Description
virtualethernet_s	SC_VirtualEthernet	Slave	-

### 3.2.38 SystemC2v7VGICConfig

Converts SystemC to v7\_vgic\_configuration\_protocol. This model is written in LISA+.

#### Iris and MTI instances for SystemC2v7VGICConfig

This model has the following Iris instances:

**Table 3-84: SystemC2v7VGICConfig Iris instances**

InstanceName	ComponentName
SystemC2v7VGICConfig	SystemC2v7VGICConfig

#### Ports for SystemC2v7VGICConfig

**Table 3-85: Ports**

Name	Protocol	Type	Description
all_received_s	AMBAPVSignal	Slave	From SystemC.
cpu_interface_number_s	AMBAPVValue64	Slave	From SystemC.
inout_cluster_number_s	AMBAPVValue64	Slave	From SystemC.
inout_cpu_number_in_cluster_s	AMBAPVValue64	Slave	From SystemC.
master_id_mask_s	AMBAPVValue	Slave	From SystemC.
master_id_s	AMBAPVValue	Slave	From SystemC.
number_of_cores_s	AMBAPVValueState	Slave	From SystemC.
response_m	AMBAPVSignal	Master	To SystemC.
v7_vgic_config_m	v7_VGIC_Configuration_Protocol	Master	v7_VGIC_Configuration_Protocol output.

### 3.2.39 VECB2AMBAPVValue64

VECB protocol to AMBA-PV protocol converter. This model is written in LISA+.

#### Iris and MTI instances for VECB2AMBAPVValue64

This model has the following Iris instances:

**Table 3-86: VECB2AMBAPVValue64 Iris instances**

InstanceName	ComponentName
VECB2AMBAPVValue64	VECB2AMBAPVValue64

### Ports for VECB2AMBAPVValue64

**Table 3-87: Ports**

Name	Protocol	Type	Description
amba_pv_ctrl_m	AMBAPVValue	Master	AMBAPV portout.
amba_pv_data_m	AMBAPVValue64	Master	AMBAPV portout.
vecb_s	VECBProtocol	Slave	VECB port in.

### 3.2.40 VirtualEthernet2SystemC

VirtualEthernet to SystemC Converter. This model is written in LISA+.

#### Iris and MTI instances for VirtualEthernet2SystemC

This model has the following Iris instances:

**Table 3-88: VirtualEthernet2SystemC Iris instances**

InstanceName	ComponentName
VirtualEthernet2SystemC	VirtualEthernet2SystemC

### Ports for VirtualEthernet2SystemC

**Table 3-89: Ports**

Name	Protocol	Type	Description
virtualethernet_m	SC_VirtualEthernet	Master	-
virtualethernet_s	VirtualEthernet	Slave	-

### 3.2.41 v7VGICConfig2SystemC

Converts v7\_vgic\_configuration\_protocol to SystemC. This model is written in LISA+.

#### Iris and MTI instances for v7VGICConfig2SystemC

This model has the following Iris instances:

**Table 3-90: v7VGICConfig2SystemC Iris instances**

InstanceName	ComponentName
v7VGICConfig2SystemC	v7VGICConfig2SystemC

### Ports for v7VGICConfig2SystemC

**Table 3-91: Ports**

Name	Protocol	Type	Description
all_received	AMBAPVSignal	Master	Called when all other values have been set in opposite bridge.
cpu_interface_number_m	AMBAPVValue64	Master	To SystemC.



Name	Protocol	Type	Description
inout_cluster_number_m	AMBAPVValue64	Master	To SystemC.
inout_cpu_number_in_cluster_m	AMBAPVValue64	Master	To SystemC.
master_id_m	AMBAPVValue	Master	To SystemC.
master_id_mask_m	AMBAPVValue	Master	To SystemC.
number_of_cores_m	AMBAPVValueState	Master	To SystemC.
response_s	AMBAPVSignal	Slave	From SystemC.
v7_vgic_config_s	v7_VGIC_Configuration_Protocol	Slave	v7_VGIC_Configuration_Protocol input.

### 3.3 Bus components

This section describes the Bus components.

PVBus is the bus protocol that is used to model all memory-like buses in Fast Models. PVBus is an internal protocol. The PVBus components, which abstract the internal details, are the interface to the PVBus API.

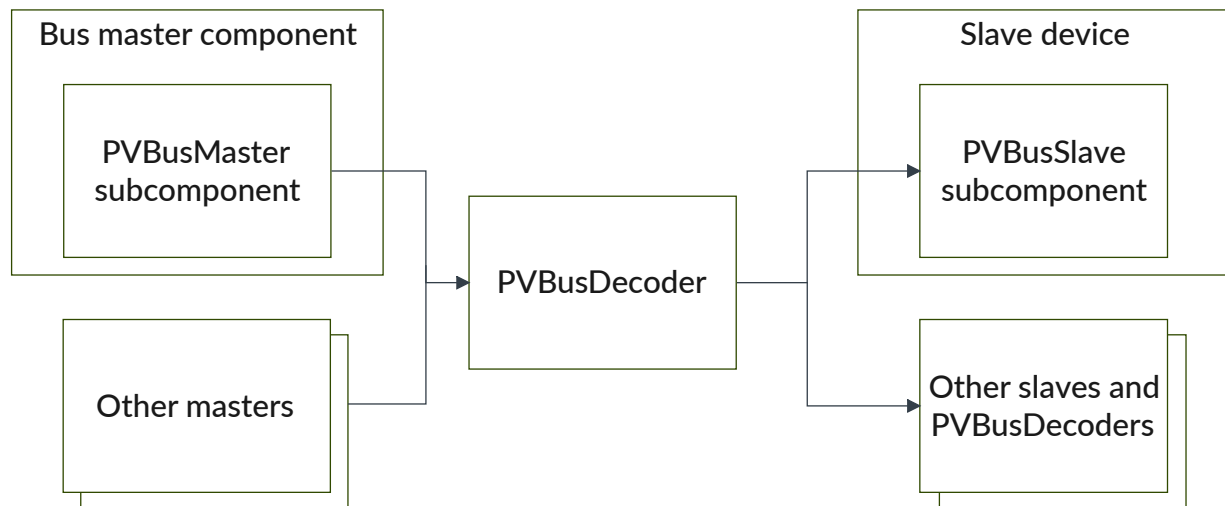
PVBus components provide functionally accurate communication between bus masters and slaves. They are not software implementations of specific hardware, but instead are abstract components that are required by the software model environment.

There is no modeling of handshaking or cycle counts. By removing this level of detail, and by using efficient internal communication mechanisms, PVBus components can provide very fast modeling of bus accesses. You must use these components correctly to achieve high simulation speeds.

Each bus master must contain a `PVBusMaster` subcomponent, and each bus slave must contain a `PVBusSlave` subcomponent. These subcomponents provide `PVBus` master and slave ports. Each `PVBus` master port can only connect to one slave, but any number of other masters can connect to the same slave. `PVBusDecoder`, `PVBusMaster` and `PVBusSlave` components communicate using the `PVBus` protocol.

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

`PVBusSlave` subcomponents provide built-in support for declaring memory-like, device-like, abort or ignore address ranges. `PVBus` has support for dealing efficiently with memory-like devices such as RAM, ROM, and Flash.

**Figure 3-1: Sample bus topology**

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

Fast Models provides some example `PVBus` systems:

- `$PVLIB_HOME/examples/LISA/common/LISA/RemapDecoder.lisa`. This example dynamically modifies routing of requests based on a remap signal, using the `TzSwitch` component.
- The directory `$PVLIB_HOME/examples/LISA/BusComponents/` contains a set of example components that show various ways of using the `PVBus` interface.

### 3.3.1 Labeller

This model is written in LISA+.

#### Changes in 11.29.19

Parameters added:

- `diagnostics`
- `use_msb_for_manager_id`

## About Labeller

Labeller and LabellerForDMA330 are utility components that allow the system designer to embed values into the Label field for transactions generated by a Bus Master. They are located between PVBUS Master and Slave ports.

As FastModels have no direct concept of AXI ID, those components that use AXI ID information have to use a proxy for it.

By default, Labeller utilizes bits [31:16] of the component's `MasterID` to store a label (see `PVMemoryAttributes.h`). Additionally, the `use_msb_for_manager_id` parameter is available, enabling the use of bits [63:48] of the `MasterID` to differentiate transactions initiated by distinct managers.

Those components that need to know an analog of AXI ID should have a configurable mapping from 'label' to its internal representation of AXI ID.

When assembling a SoC the designer has to place a labeller under every component that has to be distinguished and assign it a unique label.

The following example creates a labeller to add an ID for an HDLCD controller that is upstream of a TZC\_400. The system designer specifies a unique set of IDs for use as Non-Secure Access IDs (NSAIDs) in the TZC\_400. The labeller can insert these IDs directly into the transaction.

```
p1370_hdlcd : PL370_HDLCDC();
hdlcd_labeller : Labeller( "label" = 2 );
p1370_hdlcd.pvbus_m => hdlcd_labeller.pvbus_s;
hdlcd_labeller.pvbus_m => output_bus.pvbus_s;
```

## Iris and MTI instances for Labeller

This model has the following Iris instances:

**Table 3-92: Labeller Iris instances**

InstanceName	ComponentName
Labeller	Labeller
Labeller.pvbusmodifier	PVBusMapper

This model has the following MTI trace components:

**Table 3-93: Labeller MTI instances**

InstanceName	ComponentName
Labeller.pvbusmodifier	PVBusMapper

## Ports for Labeller

**Table 3-94: Ports**

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Output with modified MasterID.
pvbus_s	PVBus	Slave	Unmodified input.

## Parameters for Labeller

### **diagnostics**

**Type**

int

**Default value**

0x0

**Description**

Diagnostics 0=Disable, 1=Warnings, 2= Information (i.e., useful messages), 3=Debug information (i.e., all traces).

### **label**

**Type**

int

**Default value**

0x0

**Description**

The label to apply to all transactions flowing through the labeller.

### **use\_msb\_for\_manager\_id**

**Type**

bool

**Default value**

0x0

**Description**

Use bits 63:48 for unique IDs.

## 3.3.2 LabellerForDMA330

This model is written in LISA+.

### **About LabellerForDMA330**

As FastModels have no direct concept of AXI ID those components that use AXI ID information have to use a proxy for it.

We use the top 16 bits of the component's `MasterID` to store a label (see `PVMemoryAttributes.h`)

Those components that need to know an analog of AXI ID should have a configurable mapping from `label` to its internal representation of AXI ID.

When assembling a SoC the designer has to place a Labeller under every component that has to be distinguished and assign it a unique label.

This specific labeller understands the `MasterID` used by the instruction stream and uses a different label for it. It also provides the option of discriminating the DMA-330 data channels.

### Iris and MTI instances for LabellerForDMA330

This model has the following Iris instances:

**Table 3-95: LabellerForDMA330 Iris instances**

InstanceName	ComponentName
LabellerForDMA330	LabellerForDMA330
LabellerForDMA330.pvbusmapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-96: LabellerForDMA330 MTI instances**

InstanceName	ComponentName
LabellerForDMA330.pvbusmapper	PVBusMapper

### Ports for LabellerForDMA330

**Table 3-97: Ports**

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Output with modified MasterID.
pvbus_s	PVBus	Slave	Unmodified input.

### Parameters for LabellerForDMA330

#### `dma330_data_label`

##### Type

int

##### Default value

0x0

##### Description

The label to apply to all `_data_` transactions flowing through the labeller. Used as a base value in conjunction with the channel ID if data-channel discrimination is enabled.

#### `dma330_discriminate_data_channels`

##### Type

bool

##### Default value

0x0

##### Description

Discriminate between DMA-330 data channels. Channel ID is added to the data label.

**`dma330_ns_instruction_label`****Type**

int

**Default value**

0x0

**Description**

The label to apply to all non-secure `_instructions_` transactions flowing through the labeller.

**`dma330_s_instruction_label`****Type**

int

**Default value**

0x0

**Description**

The label to apply to all secure `_instructions_` transactions flowing through the labeller.

### 3.3.3 LabellerForGPUProtMode

This model is written in LISA+.

**About LabellerForGPUProtMode**

This component adds Non-Secure Access IDs (NSAIDs) to the transactions generated by the GPU. The NSAID is a four-bit number. It allows other components, such as a TrustZone Controller (TZC) or a Dynamic Memory Controller (DMC) to filter transactions and control access to memory regions that are designated as protected.

Certain Bifrost GPUs support a protected mode of operation intended to stop valuable or 'protected' data, for example the decoded frames of a DRM protected movie being written to memory that is generally accessible.

They tell the rest of the system they are in this mode by setting the signal `PROTMODE`.

External hardware outside the GPU must respond to this by making whatever adjustment is required to ensure the content goes to memory that is not generally accessible.

This labeller represents such hardware in an effort to ensure it is not forgotten in the corresponding RTL.

**Iris and MTI instances for LabellerForGPUProtMode**

This model has the following Iris instances:

**Table 3-98: LabellerForGPUProtMode Iris instances**

InstanceName	ComponentName
LabellerForGPUProtMode	LabellerForGPUProtMode
LabellerForGPUProtMode.pvbusmodifier	PVBusMapper

This model has the following MTI trace components:

**Table 3-99: LabellerForGPUProtMode MTI instances**

InstanceName	ComponentName
LabellerForGPUProtMode.pvbusmodifier	PVBusMapper

### Ports for LabellerForGPUProtMode

**Table 3-100: Ports**

Name	Protocol	Type	Description
prot_mode	Signal	Slave	Input to determine whether output is supposed to be protected or not
pvbus_m	PVBus	Master	Output with modified MasterID.
pvbus_s	PVBus	Slave	Unmodified input.

### Parameters for LabellerForGPUProtMode

#### gpu\_id\_normal

##### Type

int

##### Default value

0x0

##### Description

NSAID to apply to all transactions flowing through the labeller when prot\_mode is low.

#### gpu\_id\_protected

##### Type

int

##### Default value

0x0

##### Description

NSAID to apply to all transactions flowing through the labeller when prot\_mode is high.

## 3.3.4 MSIRewriter

Recognise writes to the GITS\_TRANSLATER register and rewrite them to go to the GITS\_TRANSLATE64R register. The DeviceID is expected to be in the bottom 32 bits of ExtendedID of the transaction. Debug transactions and reads are not rewritten. This component

can also, optionally, apply a 16 bit 'label' to the top 16 bits of the MasterID in the same way that the Labeller component does. This model is written in LISA+.

## About MSIRewriter

MSIRewriter is a component that implements the functionality of the MSI-64 Encapsulator in GIC IP, for example GIC-700. For more information about GIC-700, see [GIC-700 Technical Reference Manual](#). For the GIC architecture specification version 3 and version 4, see [GIC Architecture Specification](#).

If an MSIRewriter component is used, it converts writes to the `GITS_TRANSLATER` register to writes to a model-only register called `GITS_TRANSLATE64R`. `GITS_TRANSLATE64R` holds the DeviceID in the upper 32 bits and the EventID in the lower 32 bits. The lower 32 bits of `GITS_TRANSLATE64R` correspond to the `GITS_TRANSLATER` register.

The `GITS_TRANSLATER` register is in a page by itself except for the `GITS_TRANSLATE64R` register.

Any 16/32 bit, single beat write to `GITS_TRANSLATER` is rewritten to a 64 bit write to `GITS_TRANSLATE64R` where the top 32 bits represent the DeviceID and are assumed to come from the bottom 32 bits of ExtendedID.

## MasterID, ExtendedID, and UserFlags

These tables show how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-101: pvbus\_s port**

PVBus attribute	Bits used	Property encoded	Notes
MasterID	–	–	Not used
ExtendedID	Bits[63:32]	Stream ID	–
	Bits[31:0]	Device ID	–
UserFlags	–	–	Not used

**Table 3-102: pvbus\_m port**

PVBus attribute	Bits used	Property encoded	Notes
MasterID	Bits[31:16]	Optional 16-bit label	For usage, see the <code>label</code> parameter
ExtendedID	–	–	Not used
UserFlags	–	–	Not used

## Iris and MTI instances for MSIRewriter

This model has the following Iris instances:

**Table 3-103: MSIRewriter Iris instances**

InstanceName	ComponentName
MSIRewriter	MSIRewriter
MSIRewriter.mapper	PVBusMapper
MSIRewriter.pvbusmaster	PVBusMaster



InstanceName	ComponentName
MSIRewriter.pvbuslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-104: MSIRewriter MTI instances**

InstanceName	ComponentName
MSIRewriter	MSIRewriter
MSIRewriter.mapper	PVBusMapper
MSIRewriter.pvbusmaster	PVBusMaster
MSIRewriter.pvbuslave	PVBusSlave

### Ports for MSIRewriter

**Table 3-105: Ports**

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-

### Parameters for MSIRewriter

#### GITS\_TRANSLATE64R\_OFFSET

##### Type

int

##### Default value

0x10048

##### Description

Offset of GITS\_TRANSLATE64R from the ITS base address. When rewrite\_offset is non-zero, this option is ignored.

#### ITS0-base

##### Type

int

##### Default value

0x0

##### Description

Register base address for ITS0. This base address is used to recognise writes to the GITS\_TRANSLATER register within the ITS0's register frame. Ignored when translate\_frame\_base\_addresses is non-empty string.

**enable\_rewriting****Type**

bool

**Default value**

0x1

**Description**

Enable rewriting.

**label****Type**

int

**Default value**

0xffffffff

**Description**

If  $< 2^{*16}$  then this is a label that is put in the top 16 bits of MasterID in the same way that the component Labeller does. This labelling is not controlled by enable\_rewriting and is performed on all transactions (even rewritten ones).

**log****Type**

int

**Default value**

0x0

**Description**

Log level, 0 is off.

**rewrite\_offset****Type**

int

**Default value**

0x0

**Description**

The offset of the address where the rewritten transaction should target calculated from the original target address. When this option is non-zero, GITS\_TRANSLATE64R\_OFFSET is ignored.

**translate\_frame\_base\_addresses****Type**

string

**Default value**

""

**Description**

Comma separated list of base addresses for the 64KB translation frames of ITS instances (not the ITS base address). These addresses are used to recognise writes to the GITS\_TRANSLATER register.

### 3.3.5 PASSwitch

Allow transactions from Realm Management Extension(RME) worlds (realm/root/secure/non\_secure) to be routed separately. This model is written in C++.

**Changes in 11.29.19**

Parameters added:

- non\_secure\_protected\_port\_index
- system\_agent\_port\_index

**Iris and MTI instances for PASSwitch**

This model has the following Iris instances:

**Table 3-106: PASSwitch Iris instances**

InstanceName	ComponentName
PASSwitch	PASSwitch
PASSwitch.mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-107: PASSwitch MTI instances**

InstanceName	ComponentName
PASSwitch.mapper	PVBusMapper

**Ports for PASSwitch****Table 3-108: Ports**

Name	Protocol	Type	Description
control	PASSwitchControl	Slave	Controls routing of transactions.
pvbuse_m[6]	PVBus	Master	Manager ports of PASSwitch.
pvbuse_s	PVBus	Slave	Subordinate port of PASSwitch.

## Parameters for PASSwitch

### **non\_secure\_port\_index**

**Type**

int32\_t

**Default value**

1

Port index for Non-Secure world transactions to exit or:

**-2**

IGNORE

**-1**

ABORT

### **non\_secure\_protected\_port\_index**

**Type**

int32\_t

**Default value**

5

Port index for Non-Secure Protected world transaction to exit or:

**-2**

IGNORE

**-1**

ABORT

### **port\_map\_json**

**Type**

string

**Default value**

""

A JSON value describing ports for different address regions.

The `begin` address and `size` values should be aligned to 4KiB. The format is as follows:

```
[
  {
    "begin": 0x0,
    "size": 0x1000,
    "port": 0
  },
  {
    "begin": 0x20000,
    "size": 0x5000,
```

```
    "port": 2  
  }  
]
```

**realm\_port\_index****Type**

int32\_t

**Default value**

3

Port index for Realm world transactions to exit or:

**-2**

IGNORE

**-1**

ABORT

**root\_port\_index****Type**

int32\_t

**Default value**

2

Port index for Root world transactions to exit or:

**-2**

IGNORE

**-1**

ABORT

**secure\_port\_index****Type**

int32\_t

**Default value**

0

Port index for Secure world transactions to exit or:

**-2**

IGNORE

**-1**

ABORT

**system\_agent\_port\_index****Type**

int32\_t

**Default value**

4

Port index for system agent to exit or:

**-2**

IGNORE

**-1**

ABORT

### 3.3.6 PVBus4KBTo1KBSplitter

Takes 4KB of address range input on slave port and routes each 1KB to four different master ports. This model is written in LISA+.

**About PVBus4KBTo1KBSplitter**

The purpose of this component is to allow an upstream component to access four downstream components in the same 4 KB address range. It splits the 4 KB range from 0 to 0xfff into the following four 1 KB ranges, which allows four different components to be attached to the 4 KB range:

- 0x0-0x3ff
- 0x400-0x7ff
- 0x800-0xbff
- 0xc00-0xfff

This overcomes a limitation of PVBus which only allows components to be attached to memory addresses that are a multiple of 4 KB in size.

**Note**

The forwarded transactions have their address re-aligned with the 1 KB boundary in the range 0x0-0x3ff. For example, address 0x0402 becomes address 0x002 of the second peripheral, which is the one attached to `pvbus_m[1]`.

**Limitations**

Unaligned transactions that cross the boundaries between two peripherals are not supported. For example, when unaligned transactions are enabled by your models, you can access two double words at address 0x03ed, but you cannot access two double words at address 0x3f7.

**Iris and MTI instances for PVBus4KBTo1KBSplitter**

This model has the following Iris instances:

**Table 3-109: PVBUS4KBTo1KBSplitter Iris instances**

InstanceName	ComponentName
PVBus4KBTo1KBSplitter	PVBus4KBTo1KBSplitter
PVBus4KBTo1KBSplitter.input_slave	PVBusSlave
PVBus4KBTo1KBSplitter.output_master0	PVBusMaster
PVBus4KBTo1KBSplitter.output_master1	PVBusMaster
PVBus4KBTo1KBSplitter.output_master2	PVBusMaster
PVBus4KBTo1KBSplitter.output_master3	PVBusMaster

This model has the following MTI trace components:

**Table 3-110: PVBUS4KBTo1KBSplitter MTI instances**

InstanceName	ComponentName
PVBus4KBTo1KBSplitter.input_slave	PVBusSlave
PVBus4KBTo1KBSplitter.output_master0	PVBusMaster
PVBus4KBTo1KBSplitter.output_master1	PVBusMaster
PVBus4KBTo1KBSplitter.output_master2	PVBusMaster
PVBus4KBTo1KBSplitter.output_master3	PVBusMaster

### Ports for PVBUS4KBTo1KBSplitter

**Table 3-111: Ports**

Name	Protocol	Type	Description
pdbus_m[4]	PVBus	Master	The four downstream ports to be connected to peripherals. Each port covers 1KiB of the address space. Output address on each port will be in the range 0x0 - 0x03FF.
pdbus_s	PVBus	Slave	The upstream port. Accepts addresses in range 0x0 - 0x0FFF. Outside of this range transactions will abort.

## 3.3.7 PVBusCache

A PVBusCache manages cache-line data and supports forwarding of transactions. This model is written in C++.

### About PVBusCache

This component defines parameters and ports that are private, subject to change, and should not be used outside of the PL310 model.

### Ports for PVBusCache

**Table 3-112: Ports**

Name	Protocol	Type	Description
bus_in[4]	PVBus	Slave	Connections in from bus master.
bus_out[4]	PVBus	Master	Connections out to bus slaves.
control	PVBusCacheControl	Slave	Configuration and control port.

Name	Protocol	Type	Description
device	PVBusCacheDevice	Master	Connection out to cache device.

### 3.3.8 PVBusDecoder

A PVBusDecoder allows bus transactions to be routed to one of many slaves, based on the address given in the transaction. This model is written in C++.

#### About PVBusDecoder

Each slave connection is associated with a specific address range on the `pvbus_m_range` port. In LISA+, the syntax for this is:

```
decoder.pvbus_m_range[start..end] = slave.pvbus
```

The values for start (inclusive) and end (inclusive) must specify a 4KB-aligned region of a multiple of 4K bytes. You can specify an address range for the slave, where the decoder remaps addresses into the appropriate range. The default address range for a slave is  $[0-(\text{sizeofMasterRange} - 1)]$ .

Examples of usage:

```
component PlatformDecoder
{
    slave port<PVBus> pvbus_s;
    master port<PVBus> sdram;
    master port<PVBus> flash;
    master port<PVBus> uart;

    composition
    {
        pvdecoder : PVBusDecoder;
    }

    connection
    {
        self.pvbus_s => pvdecoder.pvbus_s;
        pvdecoder.pvbus_m_range[0x000000..0x0ffffff] => sdram;
        pvdecoder.pvbus_m_range[0x100000..0x1ffffff] => flash;
        pvdecoder.pvbus_m_range[0x200000..0x2ffffff] => uart;
        pvdecoder.pvbus_m_range[0xff0000..0xffffffff] =>
        sdram[0x070000..0x07ffff];
    }
}
```

#### Ports for PVBusDecoder

**Table 3-113: Ports**

Name	Protocol	Type	Description
pvbus_m_range	PVBus	Master	Specifies the address range for the bus master. The range must be 4KB aligned and a multiple of 4KB in size. If the address range is larger than the size of the slave device, the slave is aliased.
pvbus_s	PVBus	Slave	Accepts incoming transactions. Connect this port to a bus master, or to the output of another bus decoder.



### 3.3.9 PVBusExclusiveMonitor

Global exclusive monitor. This model is written in C++.

#### Iris and MTI instances for PVBusExclusiveMonitor

This model has the following Iris instances:

**Table 3-114: PVBusExclusiveMonitor Iris instances**

InstanceName	ComponentName
PVBusExclusiveMonitor	PVBusExclusiveMonitor
PVBusExclusiveMonitor.bus_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-115: PVBusExclusiveMonitor MTI instances**

InstanceName	ComponentName
PVBusExclusiveMonitor	PVBusExclusiveMonitor
PVBusExclusiveMonitor.bus_mapper	PVBusMapper

#### Ports for PVBusExclusiveMonitor

**Table 3-116: Ports**

Name	Protocol	Type	Description
excl_cleared	Signal	Master	Exclusive monitor clear signal port.
pdbus_m	PVBus	Master	Bus master port.
pdbus_s	PVBus	Slave	Bus slave port.

#### Parameters for PVBusExclusiveMonitor

##### **apply\_access\_width\_criteria\_to\_non\_excl\_stores**

###### Type

bool

###### Default value

0x1

###### Description

Apply the given exclusive store width matching criteria to non-exclusive stores.

##### **clear\_on\_strex\_address\_mismatch**

###### Type

bool

###### Default value

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**enable\_component****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**log2\_granule\_size****Type**

int

**Default value**

0x0

**Description**

log2 of address granule size.

**match\_access\_width****Type**

bool

**Default value**

0x0

**Description**

Fail STREX if not the same access width as LDREX.

**match\_secure\_state****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**monitor\_access\_level****Type**

uint32\_t

**Default value**

0

Which accesses to monitor:

**0**

Monitor all accesses

**1**

Monitor all accesses except WriteBack

**2**

Only monitor accesses with memory type NonCacheable or Device

**monitor\_non\_excl\_stores****Type**

bool

**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**number\_of\_monitors****Type**

int

**Default value**

0x8

**Description**

Number of monitors.

**shareability\_domain****Type**

unsigned

**Default value**

3

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored

- 0 non-shared
- 1 inner
- 2 outer
- 3 system

### 3.3.10 PVBusExclusiveSquasher

Squashes the exclusive attribute on bus transactions. This model is written in LISA+.

#### Iris and MTI instances for PVBusExclusiveSquasher

This model has the following Iris instances:

**Table 3-117: PVBusExclusiveSquasher Iris instances**

InstanceName	ComponentName
PVBusExclusiveSquasher	PVBusExclusiveSquasher
PVBusExclusiveSquasher.bus_modifier	PVBusMapper

This model has the following MTI trace components:

**Table 3-118: PVBusExclusiveSquasher MTI instances**

InstanceName	ComponentName
PVBusExclusiveSquasher.bus_modifier	PVBusMapper

#### Ports for PVBusExclusiveSquasher

**Table 3-119: Ports**

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Bus master port.
pvbus_s	PVBus	Slave	Bus slave port.

### 3.3.11 PVBusLogger

A PVBusLogger has a slave and a master port and traffic is passed straight through. All traffic is logged using an MTI trace event. This model is written in C++.

#### Iris and MTI instances for PVBusLogger

This model has the following Iris instances:

**Table 3-120: PVBusLogger Iris instances**

InstanceName	ComponentName
PVBusLogger	PVBusLogger
PVBusLogger.mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-121: PVBusLogger MTI instances**

InstanceName	ComponentName
PVBusLogger	PVBusLogger
PVBusLogger.mapper	PVBusMapper

### Ports for PVBusLogger

**Table 3-122: Ports**

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Bus master port.
pvbus_s	PVBus	Slave	Bus slave port.

### Parameters for PVBusLogger

#### **trace\_debug**

##### **Type**

bool

##### **Default value**

0x0

##### **Description**

Enable tracing of debug transactions.

#### **trace\_snoops**

##### **Type**

bool

##### **Default value**

0x0

##### **Description**

Enable tracing of ACE snoop requests.

### 3.3.12 PVBusMapper

Allow transactions to be remapped arbitrarily. This model is written in C++.

#### About PVBusMapper

This component is similar to `PVBusModifier`, but in addition:

- It has multiple downstream ports
- It allows routing of transactions to any one of these ports
- It allows arbitrary remapping of transaction addresses and attributes

As a generic modeling component, it does not have a hardware revision code.

For an example of how to use `PVBusMapper`, see `$PVLIB_HOME/examples/LISAPlus/RemappingWithPVBusMapper/`.

#### Iris and MTI instances for PVBusMapper

This model has the following Iris instances:

**Table 3-123: PVBusMapper Iris instances**

InstanceName	ComponentName
PVBusMapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-124: PVBusMapper MTI instances**

InstanceName	ComponentName
PVBusMapper	PVBusMapper

#### Ports for PVBusMapper

**Table 3-125: Ports**

Name	Protocol	Type	Description
control	PVBusMapperControl	Master	Configuration port to determine mappings.
pdbus_m[64]	PVBus	Master	Bus master ports.
pdbus_s	PVBus	Slave	Bus slave port.
reset	Signal	Slave	Reset signal.

### 3.3.13 PVBusMaster

The PVBusMaster subcomponent allows a device to generate PVBus transactions. This model is written in C++.

#### About PVBusMaster

The `PVBusMaster` subcomponent allows a device to generate PVBus transactions. It does this by providing a control port that allows a component to instantiate `pv::TransactionGenerator` objects. These objects can be used to generate bus transactions.

See `PVTransactionMasterProtocol.lisa` for details.

A bus mastering component should connect the `pvbus_m` port to its own bus port.

Example:

```
component DmaTransfer
{
    master port<PVBus> pvbus_m;
    master port<PVTransactionMaster> busmaster_control;
    composition {
        busmaster : PVBusMaster;
    }
    resources {
        pv::TransactionGenerator* stream_in;
        pv::TransactionGenerator* stream_out;
    }
    connection {
        busmaster.pvbus_m => self.pvbus_m;
        self.busmaster_control => busmaster.control;
    }
    behaviour init() {
        stream_in = busmaster_control.createTransactionGenerator();
        stream_out = busmaster_control.createTransactionGenerator();
        composition.init();
    }
    behaviour terminate() {
        delete stream_in;
        delete stream_out;
        composition.terminate();
    }
    behaviour transfer(pv::bus_addr_t start,
                      pv::bus_addr_t end,
                      pv::bus_addr_t destination)
    {
        uint32_t data;
        bool ok = true;
        while (ok && start < end) {
            ok = stream_in->read32(start, &data);
            if (ok) {
                ok = stream_out->write32(destination, &data);
            }
            start += 4;
            destination += 4;
        }
    }
}
```

#### Iris and MTI instances for PVBusMaster

This model has the following Iris instances:

**Table 3-126: PVBusMaster Iris instances**

InstanceName	ComponentName
PVBusMaster	PVBusMaster

This model has the following MTI trace components:

**Table 3-127: PVBusMaster MTI instances**

InstanceName	ComponentName
PVBusMaster	PVBusMaster

### Ports for PVBusMaster

**Table 3-128: Ports**

Name	Protocol	Type	Description
control	PVTransactionMaster	Slave	Enables the owning component to instantiate pv::TransactionGenerator objects.
pvbus_m	PVBus	Master	Sends out generated transactions to the bus.
reset	Signal	Slave	On the de-assert of this signal, a reset of the bus master will be latched this is used by the bus deadlock detection logic.

## 3.3.14 PVBusModifier

Allow transactions to be remapped arbitrarily. This model is written in C++.

### About PVBusModifier

Allow the connections to be modified through the component.

When a transaction is made to a 4 KiB address region, then the transaction is made through a channel, and if one doesn't exist then it must create one. The channel creation request is made with the specific attributes of the transaction and it is up to the system to determine where the end point of that channel should be.

This component allows you to intercept the channel creation process and change the attributes for that channel as it flows through this component.

For example, you could remap the address, or the attributes, or both.



Note

Channels are created and destroyed for any reason and so for a simulation to be deterministic then the component should always remap channels idempotently.

### Iris and MTI instances for PVBusModifier

This model has the following Iris instances:



**Table 3-129: PVBusModifier Iris instances**

InstanceName	ComponentName
PVBusModifier	PVBusMapper

This model has the following MTI trace components:

**Table 3-130: PVBusModifier MTI instances**

InstanceName	ComponentName
PVBusModifier	PVBusMapper

### Ports for PVBusModifier

**Table 3-131: Ports**

Name	Protocol	Type	Description
control	PVBusMapperControl	Master	Configuration port to determine mappings.
pvbus_m	PVBus	Master	Bus master port.
pvbus_s	PVBus	Slave	Bus slave port.
reset	Signal	Slave	Reset signal.

## 3.3.15 PVBusRouter

Allow transactions to be routed arbitrarily. This model is written in LISA+.

### Iris and MTI instances for PVBusRouter

This model has the following Iris instances:

**Table 3-132: PVBusRouter Iris instances**

InstanceName	ComponentName
PVBusRouter	PVBusRouter
PVBusRouter.mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-133: PVBusRouter MTI instances**

InstanceName	ComponentName
PVBusRouter.mapper	PVBusMapper

### Ports for PVBusRouter

**Table 3-134: Ports**

Name	Protocol	Type	Description
control	PVBusRouterControl	Master	Configuration port to determine filters.
pvbus_m[64]	PVBus	Master	Bus master ports.
pvbus_s	PVBus	Slave	Bus slave port.

### 3.3.16 PVBusSlave

A PVBusSlave handles incoming transactions, and handles support for mapping regions of device address space to work as RAM/ROM/device memory. This model is written in C++.

#### About PVBusSlave

Any component that acts as a bus slave must:

- Provide a `PVBus` slave port.
- Instantiate a PVBusSlave subcomponent, with the size parameter configured for the address range covered by the device.
- Connect the slave port to the `pvbus_s` port on the PVBusSlave.

A PVBusSlave handles incoming transactions, and handles support for mapping regions of device address space to work as RAM/ROM/device memory.

See `PVBusSlaveControlProtocol.lisa` for details of the mechanisms for configuring the memory regions.

The PVBusSlave `size` parameter controls the addressable size of the device. Addresses outside of this range will wrap around.

By default, the entire device address range is treated as device memory, meaning that all accesses will be routed to the device port. A component implementing device registers should connect the device port to a slave port that implements the `read()` and `write()` behaviors. (See the first example below).

A component that wants to implement regions of RAM or ROM must use the control port to reconfigure the PVBusSlave's decoding. See the second example below.

Example of usage:

```
component BitLatch
{
    resources
    {
        flag : bool;
    }
    slave port<PVBus> pvbus_s;

    slave port<PVDevice> device_port
    {
        behaviour read(pv::ReadTransaction tx)
        {
            if (tx.getAddress() != 0)
            {
                return tx.generateAbort();
            }
            return tx.write8(flag ? 1 : 0);
        }
        behaviour write(pv::WriteTransaction tx)
        {
            if (tx.getAddress() != 0)
            {
```

```

        return tx.generateAbort();
    }
    flag = ((tx.read8() & 1) != 0);
    return tx.writeComplete();
}

behavior debugRead(pv::ReadTransaction tx) : pv::Tx_Result
{
    return device_port.read(tx);
}

behavior debugWrite(pv::WriteTransaction tx) : pv::Tx_Result
{
    return device_port.write(tx);
}
}
composition
{
    busslave : PVBUSSlave(size=0x1000);
}
connection
{
    self.pvbus_s => busslave.pvbus_s;
    busslave.device => self.device_port;
}
}

component RAM
{
    slave port<PVBUS> pvbus_s;
    master port<PVBUSSlaveControl> busslave_control;

    composition
    {
        busslave : PVBUSSlave(size=0x01000000);
    }
    connection
    {
        self.pvbus_s => busslave.pvbus_s;
        self.busslave_control => busslave.control;
    }
    behavior init()
    {
        busslave_control.setAccess(0, 0x01000000, pv::RW, pv::MEMORY);
    }
}
}

```

## Iris and MTI instances for PVBUSSlave

This model has the following Iris instances:

**Table 3-135: PVBUSSlave Iris instances**

InstanceName	ComponentName
PVBUSSlave	PVBUSSlave

This model has the following MTI trace components:

**Table 3-136: PVBUSSlave MTI instances**

InstanceName	ComponentName
PVBUSSlave	PVBUSSlave

## Ports for PVBusSlave

**Table 3-137: Ports**

Name	Protocol	Type	Description
control	<a href="#">PVBusSlaveControl</a>	Slave	Enables the owning component to control which regions of the device memory are to be handled as RAM/ROM/Device. These settings can be changed dynamically. For example, when a Flash component is being programmed, it can switch to treating reads as Device requests instead of ROM requests.
device	<a href="#">PVDevice</a>	Master	Passes on requests for peripheral register accesses to permit the owning component to handle the request.
pvbus_s	<a href="#">PVBus</a>	Slave	Handles incoming requests from bus masters.
reset	<a href="#">Signal</a>	Slave	On the assert of this signal, a reset of the bus slave will be latched this is used by the bus deadlock detection logic.

### 3.3.17 PVCoherentInterconnect

PVCoherentInterconnect is a generic interconnect implementation. This model is written in LISA+.

#### About PVCoherentInterconnect

PVCoherentInterconnect is a component written in LISA+ that is designed to be a generic interconnect.

PVCoherentInterconnect supports up to 128 clusters and requires minimal configuration, greatly simplifying the effort needed to set up the interconnect for systems with a high cluster count. It can also be modified to extend the number of clusters that can be connected to 4096. It has a single downstream port to handle the bus traffic.

You can find how to use this component in the [Fast Models Tutorials](#)

#### Iris and MTI instances for PVCoherentInterconnect

This model has the following Iris instances:

**Table 3-138: PVCoherentInterconnect Iris instances**

InstanceName	ComponentName
PVCoherentInterconnect	PVCoherentInterconnect
PVCoherentInterconnect.pvcache0	PVCache64
PVCoherentInterconnect.pvcache0.upstream[0]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[10]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[11]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[12]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[13]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[14]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[15]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[16]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[17]	PVBusSlave

InstanceName	ComponentName
PVCoherentInterconnect.pvcache0.upstream[18]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[19]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[1]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[20]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[21]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[22]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[23]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[24]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[25]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[26]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[27]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[28]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[29]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[2]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[30]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[31]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[32]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[33]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[34]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[35]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[36]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[37]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[38]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[39]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[3]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[40]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[41]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[42]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[43]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[44]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[45]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[46]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[47]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[48]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[49]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[4]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[50]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[51]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[52]	PVBusSlave

InstanceName	ComponentName
PVCoherentInterconnect.pvcache0.upstream[53]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[54]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[55]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[56]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[57]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[58]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[59]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[5]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[60]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[61]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[62]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[63]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[6]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[7]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[8]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[9]	PVBusSlave
PVCoherentInterconnect.pvcache1	PVCache64
PVCoherentInterconnect.pvcache1.upstream[0]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[10]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[11]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[12]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[13]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[14]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[15]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[16]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[17]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[18]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[19]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[1]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[20]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[21]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[22]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[23]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[24]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[25]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[26]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[27]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[28]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[29]	PVBusSlave

InstanceName	ComponentName
PVCoherentInterconnect.pvcache1.upstream[2]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[30]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[31]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[32]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[33]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[34]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[35]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[36]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[37]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[38]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[39]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[3]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[40]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[41]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[42]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[43]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[44]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[45]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[46]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[47]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[48]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[49]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[4]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[50]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[51]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[52]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[53]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[54]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[55]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[56]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[57]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[58]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[59]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[5]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[60]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[61]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[62]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[63]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[6]	PVBusSlave

InstanceName	ComponentName
PVCoherentInterconnect.pvcache1.upstream[7]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[8]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[9]	PVBusSlave
PVCoherentInterconnect.pvcache_common	PVCache64
PVCoherentInterconnect.pvcache_common.upstream[0]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[10]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[11]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[12]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[13]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[14]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[15]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[16]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[17]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[18]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[19]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[1]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[20]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[21]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[22]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[23]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[24]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[25]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[26]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[27]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[28]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[29]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[2]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[30]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[31]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[32]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[33]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[34]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[35]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[36]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[37]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[38]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[39]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[3]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[40]	PVBusSlave



InstanceName	ComponentName
PVCoherentInterconnect.pvcache_common.upstream[41]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[42]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[43]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[44]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[45]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[46]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[47]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[48]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[49]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[4]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[50]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[51]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[52]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[53]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[54]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[55]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[56]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[57]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[58]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[59]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[5]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[60]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[61]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[62]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[63]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[6]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[7]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[8]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[9]	PVBusSlave

This model has the following MTI trace components:

**Table 3-139: PVCoherentInterconnect MTI instances**

InstanceName	ComponentName
PVCoherentInterconnect.pvcache0	PVCache64
PVCoherentInterconnect.pvcache0.upstream[0]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[10]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[11]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[12]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[13]	PVBusSlave

InstanceName	ComponentName
PVCoherentInterconnect.pvcache0.upstream[14]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[15]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[16]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[17]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[18]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[19]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[1]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[20]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[21]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[22]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[23]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[24]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[25]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[26]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[27]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[28]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[29]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[2]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[30]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[31]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[32]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[33]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[34]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[35]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[36]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[37]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[38]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[39]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[3]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[40]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[41]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[42]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[43]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[44]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[45]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[46]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[47]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[48]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[49]	PVBusSlave

InstanceName	ComponentName
PVCoherentInterconnect.pvcache0.upstream[4]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[50]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[51]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[52]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[53]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[54]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[55]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[56]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[57]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[58]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[59]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[5]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[60]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[61]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[62]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[63]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[6]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[7]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[8]	PVBusSlave
PVCoherentInterconnect.pvcache0.upstream[9]	PVBusSlave
PVCoherentInterconnect.pvcache1	PVCache64
PVCoherentInterconnect.pvcache1.upstream[0]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[10]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[11]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[12]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[13]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[14]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[15]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[16]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[17]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[18]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[19]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[1]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[20]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[21]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[22]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[23]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[24]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[25]	PVBusSlave

InstanceName	ComponentName
PVCoherentInterconnect.pvcache1.upstream[26]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[27]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[28]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[29]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[2]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[30]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[31]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[32]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[33]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[34]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[35]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[36]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[37]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[38]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[39]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[3]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[40]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[41]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[42]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[43]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[44]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[45]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[46]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[47]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[48]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[49]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[4]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[50]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[51]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[52]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[53]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[54]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[55]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[56]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[57]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[58]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[59]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[5]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[60]	PVBusSlave

InstanceName	ComponentName
PVCoherentInterconnect.pvcache1.upstream[61]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[62]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[63]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[6]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[7]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[8]	PVBusSlave
PVCoherentInterconnect.pvcache1.upstream[9]	PVBusSlave
PVCoherentInterconnect.pvcache_common	PVCache64
PVCoherentInterconnect.pvcache_common.upstream[0]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[10]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[11]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[12]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[13]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[14]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[15]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[16]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[17]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[18]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[19]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[1]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[20]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[21]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[22]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[23]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[24]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[25]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[26]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[27]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[28]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[29]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[2]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[30]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[31]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[32]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[33]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[34]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[35]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[36]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[37]	PVBusSlave

InstanceName	ComponentName
PVCoherentInterconnect.pvcache_common.upstream[38]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[39]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[3]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[40]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[41]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[42]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[43]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[44]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[45]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[46]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[47]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[48]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[49]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[4]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[50]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[51]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[52]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[53]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[54]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[55]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[56]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[57]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[58]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[59]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[5]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[60]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[61]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[62]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[63]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[6]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[7]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[8]	PVBusSlave
PVCoherentInterconnect.pvcache_common.upstream[9]	PVBusSlave

## Ports for PVCoherentInterconnect

**Table 3-140: Ports**

Name	Protocol	Type	Description
downstream	PVBus	Master	-
upstream[128]	PVBus	Slave	-

## Parameters for PVCoherentInterconnect

### **cache\_state\_modelled**

#### Type

bool

#### Default value

0x1

#### Description

Model the cache state to enable coherency in the interconnect. All the upstream components should have their cache state modelling on, for this to be on.

### **pvcache0.no\_distinction\_between\_ish\_and\_osh**

#### Type

bool

#### Default value

0x0

#### Description

Whether the cache treats the distinction between Inner-Shareability and Outer-Shareability as meaningful when matching attributes. Note that the Outer/Inner Shareability distinction is preserved on the bus.

### **pvcache1.no\_distinction\_between\_ish\_and\_osh**

#### Type

bool

#### Default value

0x0

#### Description

Whether the cache treats the distinction between Inner-Shareability and Outer-Shareability as meaningful when matching attributes. Note that the Outer/Inner Shareability distinction is preserved on the bus.

### **pvcache\_common.no\_distinction\_between\_ish\_and\_osh**

#### Type

bool

#### Default value

0x0

#### Description

Whether the cache treats the distinction between Inner-Shareability and Outer-Shareability as meaningful when matching attributes. Note that the Outer/Inner Shareability distinction is preserved on the bus.

### 3.3.18 PVMemoryProtectionEngine

Encrypt memory transactions for each encryption context with an independent key to prevent mismatch access. This model is written in C++.

#### About PVMemoryProtectionEngine

PVMemoryProtectionEngine is a simplified implementation of a Memory Protection Engine (MPE) component as described in [Arm Realm Management Extension \(RME\) System Architecture](#).

PVMemoryProtectionEngine supports the following features:

- Memory encryption.
- Each 4KiB page in memory is encrypted based on an encryption key. Each Physical Address Space (PAS) has a separate encryption key.
- Two or more encryption keys can be the same value.
- Configurable encryption keys for each PAS.
- Configurable encryption block size.
- Configurable corruption strategy. You can control the behavior of memory contents that are not written by the access within the encryption block.
- Encryption/decryption algorithm is a simple XOR of data with the corresponding encryption key.
- Downstream memory is always stored as plain text, allowing debuggers to view data.

For example, if a block is currently encrypted by the ns-PAS and then a byte is written by the rl-PAS, if the `block_size_in_bytes` is 4KiB, the rest of the data in the 4KiB page is corrupted such that even if you read a different byte back through the ns-PAS, you would not get the original data.

The primary use case for this component is to identify software mis-programming, where the same Physical address is accessed through more than one PAS. With PVMemoryProtectionEngine enabled, a PE sees encrypted or corrupted data when it is accessed using a different PAS to the original PAS that wrote to that page in memory.

The PVMemoryProtectionEngine component is expected to be connected in a platform at the Point of Physical Aliasing (PoPA) if storage is shared, otherwise before each specific storage for a subset of the PASes.

PVMemoryProtectionEngine imposes a runtime cost when enabled. Normally, it is only needed when debugging and verifying the Realm Management Monitor (RMM) software. If the RMM software is correct, memory contents encrypted with the wrong key would not be visible.

The PVMemoryProtectionEngine does not encrypt or corrupt the tag data for MTE, but this feature will be supported in future.

#### Iris and MTI instances for PVMemoryProtectionEngine

This model has the following Iris instances:



**Table 3-141: PVMemoryProtectionEngine Iris instances**

InstanceName	ComponentName
PVMemoryProtectionEngine	PVMemoryProtectionEngine
PVMemoryProtectionEngine.mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-142: PVMemoryProtectionEngine MTI instances**

InstanceName	ComponentName
PVMemoryProtectionEngine	PVMemoryProtectionEngine
PVMemoryProtectionEngine.mapper	PVBusMapper

### Ports for PVMemoryProtectionEngine

**Table 3-143: Ports**

Name	Protocol	Type	Description
pdbus_m	PVBus	Master	Manager ports of the MPE
pdbus_s	PVBus	Slave	Subordinate port of the MPE

### Parameters for PVMemoryProtectionEngine

#### **block\_size\_in\_bytes**

##### Type

int

##### Default value

0x1000

##### Description

Encryption block size, supported sizes are 1 or 4096.

#### **corruption\_strategy**

##### Type

uint8\_t

##### Default value

0

Corruption strategy:

0

fill with constants per-old-encryption-context

1

fill with constants per-new-encryption-context

2

random data

**enable****Type**

bool

**Default value**

0x0

**Description**

Enables the Memory Protection Engine.

**ignore\_mecid****Type**

bool

**Default value**

0x0

**Description**

Ignore MECID during encryption key calculation.

**non\_secure\_pas\_enc\_key****Type**

int

**Default value**

0x22

**Description**

Non-secure PAS encryption key.

**output\_attributes\_parameter\_of\_core****Type**

string

**Default value**

"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID,  
ExtendedID[38]=MPAM\_NS"

**Description**

Encoding of various attributes on the bus.

**realm\_pas\_enc\_key****Type**

int

**Default value**

0x88

**Description**  
Realm PAS encryption key.

**root\_pas\_enc\_key**

**Type**  
int

**Default value**  
0x44

**Description**  
Root PAS encryption key.

**secure\_pas\_enc\_key**

**Type**  
int

**Default value**  
0x11

**Description**  
Secure PAS encryption key.

3.3.19 PVWriteBuffer

The PVWriteBuffer subcomponent buffers PVBus transactions. This model is written in C++.

Iris and MTI instances for PVWriteBuffer

This model has the following Iris instances:

Table 3-144: PVWriteBuffer Iris instances

InstanceName	ComponentName
PVWriteBuffer	PVWriteBuffer
PVWriteBuffer.mapper	PVBusMapper

This model has the following MTI trace components:

Table 3-145: PVWriteBuffer MTI instances

InstanceName	ComponentName
PVWriteBuffer	PVWriteBuffer
PVWriteBuffer.mapper	PVBusMapper

## Ports for PVWriteBuffer

**Table 3-146: Ports**

Name	Protocol	Type	Description
barrier_notify_s	<a href="#">PVWriteBuffer_BarrierPort</a>	Slave	Barrier notification input.
clk_in	<a href="#">ClockSignal</a>	Slave	Clock input.
pvbus_m	<a href="#">PVBus</a>	Master	Master connection to memory bus.
pvbus_s	<a href="#">PVBus</a>	Slave	Slave connection for transactions to be buffered.
reset_in	<a href="#">Signal</a>	Slave	Reset input.
serror_notify_m	<a href="#">PVWriteBuffer_SErrorPort</a>	Master	SError output generation.

## 3.3.20 SimplePVBusMaster

Component to generate PVTransactions with configurable attributes and address. This model is written in LISA+.

### Iris and MTI instances for SimplePVBusMaster

This model has the following Iris instances:

**Table 3-147: SimplePVBusMaster Iris instances**

InstanceName	ComponentName
SimplePVBusMaster	SimplePVBusMaster
SimplePVBusMaster.clocktimer64	ClockTimerThread64
SimplePVBusMaster.clocktimer64.thread	SchedulerThread
SimplePVBusMaster.clocktimer64.thread_event	SchedulerThreadEvent
SimplePVBusMaster.pvbusmaster	PVBusMaster
SimplePVBusMaster.pvbusslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-148: SimplePVBusMaster MTI instances**

InstanceName	ComponentName
SimplePVBusMaster.pvbusmaster	<a href="#">PVBusMaster</a>
SimplePVBusMaster.pvbusslave	<a href="#">PVBusSlave</a>

## Ports for SimplePVBusMaster

**Table 3-149: Ports**

Name	Protocol	Type	Description
pvbus_m	<a href="#">PVBus</a>	Master	Output of generated transactions.
pvbus_s	<a href="#">PVBus</a>	Slave	-

## Parameters for SimplePVBUSMaster

### **verbose**

#### Type

bool

#### Default value

0x0

#### Description

verbose.

## 3.3.21 TZSwitch

Allow TrustZone secure/normal bus signals to be routed separately. This model is written in LISA+.

### About TZSwitch

Transactions received on the `pvbush_input` slave port are routed according to a configuration that is set up using parameters and/or the control port. Separate rules can be given for secure and for normal transactions.

Transactions can be routed to one of the two master ports, `pvbush_port_a` or `pvbush_port_b`, can be ignored, or can generate aborts.

The control port behaviour `routeAccesses()` takes two arguments:

- `input` selects which types of signals are reconfigured:

#### **TZINPUT\_SECURE**

Change the routing for secure transactions

#### **TZINPUT\_NORMAL**

Change the routing for normal transactions

#### **TZINPUT\_ANY**

Change the routing for all transactions

- `destination` selects how the chosen transactions are routed:

#### **TZROUTE\_IGNORE**

Transactions are ignored. Reads return 0.

#### **TZROUTE\_TO\_PORT\_A**

Route transactions to `pvbush_port_a`.

#### **TZROUTE\_TO\_PORT\_B**

Route transactions to `pvbush_port_b`.

#### **TZROUTE\_ABORT**

Cause transactions to generate an abort.

Initial routing is configured using parameters `secure` and `normal` based on the following values:

- 0
- Ignore these transactions.
- 1
- Forward the transactions to `pvbus_port_a`.
- 2
- Forward the transactions to `pvbus_port_b`.
- 3
- Generate an abort for these transactions.

Both default and explicit parameter values are overridden by any runtime calls to `routeAccesses()` on the control port.

Iris and MTI instances for TZSwitch

This model has the following Iris instances:

Table 3-150: TZSwitch Iris instances

InstanceName	ComponentName
TZSwitch	TZSwitch
TZSwitch.pvbus_mapper	PVBusMapper

This model has the following MTI trace components:

Table 3-151: TZSwitch MTI instances

InstanceName	ComponentName
TZSwitch.pvbus_mapper	PVBusMapper

Ports for TZSwitch

Table 3-152: Ports

Name	Protocol	Type	Description
control	<a href="#">TZSwitchControl</a>	Slave	Controls routing of transactions.
pvbus_input	<a href="#">PVBus</a>	Slave	Slave port for connection to PVBus master/decoder.
pvbus_port_a	<a href="#">PVBus</a>	Master	Output port a.
pvbus_port_b	<a href="#">PVBus</a>	Master	Output port b.

Parameters for TZSwitch

**normal**

Type

int

Default value

0x2

**Description**

Normal Port.

**secure****Type**

int

**Default value**

0x1

**Description**

Secure Port.

## 3.4 Clocking components

This section describes the Clocking components.

The clocking components and protocols provide a mechanism for systems to regulate the execution rate of components. Clocking includes the concept of clock rates, dividers to change clock rates, and timers to generate callbacks based on those clock rates.

If the MasterClock component is instantiated in a system, it provides a consistent master clock rate. Although this rate is not defined, you can consider this to be 1Hz, even for non-SystemC systems. ClockDivider components are able to convert this clock rate into a new rate using a multiplier and divider, although the clock rate cannot be divided to be less than 1Hz. You can cascade ClockDivider components to produce many different clock rates within a system. The maximum ratio of any two clocks in the system must be less than  $2^{32}$ .

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


### 3.4.1 ClockDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters. This model is written in C++.

#### About ClockDivider

This component uses a configurable ratio to convert the `clockSignal` rate at its input to a new `clockSignal` rate at its output. Changes to the input rate or ratio take effect immediately and clocking components dependent on the output rate continue counting at the new rate.

This component does not normally incur a runtime performance cost. However, reprogramming the clock rate causes all related clocks and timers to be recalculated.



Note

MasterClock is a 1 Hz clock. If the CPU clock frequency is not set to a realistic value, unpredictable behavior might occur, for example the simulation might freeze.

Iris and MTI instances for ClockDivider

This model has the following Iris instances:

Table 3-153: ClockDivider Iris instances

InstanceName	ComponentName
ClockDivider	ClockDivider

This model has the following MTI trace components:

Table 3-154: ClockDivider MTI instances

InstanceName	ComponentName
ClockDivider	ClockDivider

Ports for ClockDivider

Table 3-155: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Input clock signal, coming from a MasterClock or another ClockDivider.
clk_out	ClockSignal	Master	Clock signal generated by this ClockDivider.
rate	ClockRateControl	Slave	Permits you to dynamically change the clock divider ratio.

Parameters for ClockDivider

div

Type

int

Default value

0x1

Description

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

mul

Type

int



**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

### 3.4.2 ClockGate

Clock gate for dis/enabling the clock. This model is written in C++.

**Iris and MTI instances for ClockGate**

This model has the following Iris instances:

**Table 3-156: ClockGate Iris instances**

InstanceName	ComponentName
ClockGate	ClockGate
ClockGate.divider	ClockDivider

This model has the following MTI trace components:

**Table 3-157: ClockGate MTI instances**

InstanceName	ComponentName
ClockGate.divider	ClockDivider

**Ports for ClockGate****Table 3-158: Ports**

Name	Protocol	Type	Description
clk_enable	Signal	Slave	-
clk_in	ClockSignal	Slave	-
clk_out	ClockSignal	Master	-
halt	Signal	Master	-

**Parameters for ClockGate****diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

**divider.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**divider.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

### 3.4.3 ClockSelector

ClockSignal Selector. This model is written in C++.

#### Iris and MTI instances for ClockSelector

This model has the following Iris instances:

**Table 3-159: ClockSelector Iris instances**

InstanceName	ComponentName
ClockSelector	ClockSelector
ClockSelector.clkdiv0	ClockDivider
ClockSelector.clkdiv1	ClockDivider
ClockSelector.clkdiv10	ClockDivider
ClockSelector.clkdiv2	ClockDivider
ClockSelector.clkdiv3	ClockDivider
ClockSelector.clkdiv4	ClockDivider
ClockSelector.clkdiv5	ClockDivider
ClockSelector.clkdiv6	ClockDivider
ClockSelector.clkdiv7	ClockDivider
ClockSelector.clkdiv8	ClockDivider
ClockSelector.clkdiv9	ClockDivider
ClockSelector.clkdivider	ClockDivider

This model has the following MTI trace components:

**Table 3-160: ClockSelector MTI instances**

InstanceName	ComponentName
ClockSelector.clkdiv0	ClockDivider
ClockSelector.clkdiv1	ClockDivider
ClockSelector.clkdiv10	ClockDivider
ClockSelector.clkdiv2	ClockDivider
ClockSelector.clkdiv3	ClockDivider
ClockSelector.clkdiv4	ClockDivider
ClockSelector.clkdiv5	ClockDivider
ClockSelector.clkdiv6	ClockDivider
ClockSelector.clkdiv7	ClockDivider
ClockSelector.clkdiv8	ClockDivider
ClockSelector.clkdiv9	ClockDivider
ClockSelector.clkdivider	ClockDivider

### Ports for ClockSelector

**Table 3-161: Ports**

Name	Protocol	Type	Description
clk_in[11]	ClockSignal	Slave	-
clk_out	ClockSignal	Master	-
clk_sel	Signal	Slave	-
clk_sel_num	Value	Slave	-

### Parameters for ClockSelector

#### **clkdiv0.div**

##### Type

int

##### Default value

0x1

##### Description

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

#### **clkdiv0.mul**

##### Type

int

##### Default value

0x1

##### Description

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv1.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv1.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv10.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv10.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv2.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv2.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv3.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv3.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv4.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv4.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv5.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv5.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv6.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv6.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv7.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv7.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv8.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv8.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv9.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv9.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdivider.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdivider.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

### 3.4.4 ClockTimer

A ClockTimer provides support for counting a number of ticks at the rate determined by the input clock. This model is written in C++.

#### About ClockTimer

When the countdown reaches zero, the timer can send a signal to another component. That component can return a value that causes the timer to start counting down again.

Setting up a timer is very efficient, and no host processing time is used while a counter is counting down: when a timer is started, the scheduler precomputes the finish time.

See `ClockSignalProtocol.lisa` and `CounterDivider.lisa` for more details of the scheduler system. See `TimerControlProtocol.lisa` and `TimerCallbackProtocol.lisa` for the methods used to control a timer and to handle the signal when the countdown is complete.

To use a ClockTimer:



1. Connect the ClockTimer's `timer_callback` port to a slave port that implements the `signal()` behaviour.
2. Connect a clock signal to the `clk_in` port.
3. Use the `timer_control` port to start the timer counting down for a given number of ticks.

### Ports for ClockTimer

**Table 3-162: Ports**

Name	Protocol	Type	Description
<code>clk_in</code>	<a href="#">ClockSignal</a>	Slave	Determines the tick rate of the timer.
<code>timer_callback</code>	<a href="#">TimerCallback</a>	Master	Port on which a signal is sent after the number of scheduled ticks has elapsed.
<code>timer_control</code>	<a href="#">TimerControl</a>	Slave	Permits the timer to be set, canceled and queried.

## 3.4.5 ClockTimer64

A ClockTimer64 provides support for counting a number of ticks at the rate determined by the input clock. This model is written in C++.

### About ClockTimer64

When the countdown reaches zero, the timer can send a signal to another component. That component can return a value that causes the timer to start counting down again.

Setting up a timer is very efficient, and no host processing time is used while a counter is counting down. When a timer is started, the scheduler precomputes the finish time.

This version of the timer provides 64-bit resolution.

See [ClockSignalProtocol64.lisa](#) and [CounterDivider.lisa](#) for more details of the scheduler system. See [TimerControlProtocol64.lisa](#) and [TimerCallbackProtocol.lisa](#) for the methods used to control a timer and to handle the signal when the countdown is complete.

To use a ClockTimer64:

1. Connect the ClockTimer64's `timer_callback` port to a slave port that implements the `signal()` behaviour.
2. Connect a clock signal to the `clk_in` port.
3. Use the `timer_control` port to start the timer counting down for a given number of ticks.

### Ports for ClockTimer64

**Table 3-163: Ports**

Name	Protocol	Type	Description
<code>clk_in</code>	<a href="#">ClockSignal</a>	Slave	Determines the tick rate of the timer.
<code>timer_callback</code>	<a href="#">TimerCallback64</a>	Master	Port on which a signal is sent after the number of scheduled ticks has elapsed.
<code>timer_control</code>	<a href="#">TimerControl64</a>	Slave	Permits the timer to be set, canceled and queried.

### 3.4.6 ClockTimerThread

A `ClockTimerThread(64)` is a drop-in replacement for a `ClockTimer(64)` component. This model is written in LISA+.

#### About ClockTimerThread

A `ClockTimerThread(64)` is a drop-in replacement for a `ClockTimer(64)` component. The main difference to the `clockTimer` component is that the `clockTimerThread` runs the `signal()` callback from a proper scheduler thread. This means that the `signal()` function may directly or indirectly invoke `wait()` functions to wait for time or events. This is not allowed for the `clockTimer` component which does not use a thread.

Components that issue bus transactions from within the timer `signal()` callback must use `ClockTimerThread(64)` rather than `ClockTimer(64)`.

#### Iris and MTI instances for ClockTimerThread

This model has the following Iris instances:

**Table 3-164: ClockTimerThread Iris instances**

InstanceName	ComponentName
<code>ClockTimerThread</code>	<code>ClockTimerThread</code>
<code>ClockTimerThread.timer</code>	<code>ClockTimerThread64</code>
<code>ClockTimerThread.timer.thread</code>	<code>SchedulerThread</code>
<code>ClockTimerThread.timer.thread_event</code>	<code>SchedulerThreadEvent</code>

#### Ports for ClockTimerThread

**Table 3-165: Ports**

Name	Protocol	Type	Description
<code>clk_in</code>	<a href="#">ClockSignal</a>	Slave	Determines the tick rate of the timer.
<code>timer_callback</code>	<a href="#">TimerCallback</a>	Master	Port on which a signal is sent after the number of scheduled ticks has elapsed.
<code>timer_control</code>	<a href="#">TimerControl</a>	Slave	Permits the timer to be set, canceled and queried.

### 3.4.7 ClockTimerThread64

A `ClockTimerThread64` is a drop-in replacement for `ClockTimer64`. This model is written in LISA+.

#### About ClockTimerThread64

A `ClockTimerThread(64)` is a drop-in replacement for a `ClockTimer(64)` component. The main difference to the `clockTimer` component is that the `clockTimerThread` runs the `signal()` callback from a proper scheduler thread. This means that the `signal()` function may directly or indirectly invoke `wait()` functions to wait for time or events. This is not allowed for the `clockTimer` component which does not use a thread.

Components that issue bus transactions from within the timer `signal()` callback must use `ClockTimerThread(64)` rather than `ClockTimer(64)`.

### Iris and MTI instances for ClockTimerThread64

This model has the following Iris instances:

**Table 3-166: ClockTimerThread64 Iris instances**

InstanceName	ComponentName
ClockTimerThread64	ClockTimerThread64
ClockTimerThread64.thread	SchedulerThread
ClockTimerThread64.thread_event	SchedulerThreadEvent

### Ports for ClockTimerThread64

**Table 3-167: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	Determines the tick rate of the timer.
timer_callback	<a href="#">TimerCallback64</a>	Master	Port on which a signal is sent after the number of scheduled ticks has elapsed.
timer_control	<a href="#">TimerControl64</a>	Slave	Permits the timer to be set, canceled and queried.

## 3.4.8 ClusterClockControl

Cluster clock control allows input selection, rate control and gating. This model is written in C++.

### Iris and MTI instances for ClusterClockControl

This model has the following Iris instances:

**Table 3-168: ClusterClockControl Iris instances**

InstanceName	ComponentName
ClusterClockControl	ClusterClockControl
ClusterClockControl.clkGate	ClockGate
ClusterClockControl.clkGate.divider	ClockDivider
ClusterClockControl.clkSelector	ClockSelector
ClusterClockControl.clkSelector.clkdiv0	ClockDivider
ClusterClockControl.clkSelector.clkdiv1	ClockDivider
ClusterClockControl.clkSelector.clkdiv10	ClockDivider
ClusterClockControl.clkSelector.clkdiv2	ClockDivider
ClusterClockControl.clkSelector.clkdiv3	ClockDivider
ClusterClockControl.clkSelector.clkdiv4	ClockDivider
ClusterClockControl.clkSelector.clkdiv5	ClockDivider
ClusterClockControl.clkSelector.clkdiv6	ClockDivider
ClusterClockControl.clkSelector.clkdiv7	ClockDivider

InstanceName	ComponentName
ClusterClockControl.clkSelector.clkdiv8	ClockDivider
ClusterClockControl.clkSelector.clkdiv9	ClockDivider
ClusterClockControl.clkSelector.clkdivider	ClockDivider
ClusterClockControl.refClkDiv	ClockDivider
ClusterClockControl.sysClkDiv	ClockDivider
ClusterClockControl.xClkDiv	ClockDivider

This model has the following MTI trace components:

**Table 3-169: ClusterClockControl MTI instances**

InstanceName	ComponentName
ClusterClockControl.clkGate.divider	ClockDivider
ClusterClockControl.clkSelector.clkdiv0	ClockDivider
ClusterClockControl.clkSelector.clkdiv1	ClockDivider
ClusterClockControl.clkSelector.clkdiv10	ClockDivider
ClusterClockControl.clkSelector.clkdiv2	ClockDivider
ClusterClockControl.clkSelector.clkdiv3	ClockDivider
ClusterClockControl.clkSelector.clkdiv4	ClockDivider
ClusterClockControl.clkSelector.clkdiv5	ClockDivider
ClusterClockControl.clkSelector.clkdiv6	ClockDivider
ClusterClockControl.clkSelector.clkdiv7	ClockDivider
ClusterClockControl.clkSelector.clkdiv8	ClockDivider
ClusterClockControl.clkSelector.clkdiv9	ClockDivider
ClusterClockControl.clkSelector.clkdivider	ClockDivider
ClusterClockControl.refClkDiv	ClockDivider
ClusterClockControl.sysClkDiv	ClockDivider
ClusterClockControl.xClkDiv	ClockDivider

## Ports for ClusterClockControl

**Table 3-170: Ports**

Name	Protocol	Type	Description
clk_out	ClockSignal	Master	-
clkDivExt	ClockRateControl	Slave	-
clkDivSys	ClockRateControl	Slave	-
clkEnable	Signal	Slave	-
clkSel	Value	Slave	-
halt	Signal	Master	-
refClk_in	ClockSignal	Slave	-
sysClk_in	ClockSignal	Slave	-
xClk_in	ClockSignal	Slave	-

## Parameters for ClusterClockControl

### **clkGate.diagnostics**

**Type**

int

**Default value**

0x0

**Description**

Diagnostics.

### **clkGate.divider.div**

**Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

### **clkGate.divider.mul**

**Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

### **clkSelector.clkdiv0.div**

**Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

### **clkSelector.clkdiv0.mul**

**Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv1.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv1.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv10.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv10.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv2.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv2.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv3.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv3.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv4.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv4.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv5.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv5.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv6.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv6.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv7.div****Type**

int

**Default value**

0x1



**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv7.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv8.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv8.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv9.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdiv9.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdivider.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.clkdivider.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelector.diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

**refClkDiv.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**refClkDiv.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**sysClkDiv.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**sysClkDiv.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**xClkDiv.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**xClkDiv.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

### 3.4.9 MasterClock

A source of a clock signal representing the base clock rate of the simulation (nominally 1Hz). The signal from the output port can be connected to a ClockDivider, to generate clock signals at a different clock rate. The output can also be connected to a ClockTimer in order to generate

scheduled events, or to any other components that accept a `ClockSignal` input. See `ClockSignal.lisa` for more information. This model is written in C++.

### About MasterClock

This component provides a single `clockSignal` output that can be used to drive the `clockSignal` input of `clockDividers`, `clockTimers` and other clocking components.

The rate of the `MasterClock` is not defined because all clocking is relative, but can be considered to be 1 Hz.



If the CPU clock frequency is not set to a realistic value, unpredictable behavior might occur, for example the simulation might freeze.

A system might contain more than one `MasterClock`, all of which generate the same `clockSignal` rate.

### Ports for MasterClock

**Table 3-171: Ports**

Name	Protocol	Type	Description
<code>clk_out</code>	<code>ClockSignal</code>	Master	Master clock rate.

## 3.4.10 PLLControl

Simulate PLL clock frequency control logic. This model is written in C++.

### Iris and MTI instances for PLLControl

This model has the following Iris instances:

**Table 3-172: PLLControl Iris instances**

InstanceName	ComponentName
<code>PLLControl</code>	<code>PLLControl</code>
<code>PLLControl.clkdiv</code>	<code>ClockDivider</code>

This model has the following MTI trace components:

**Table 3-173: PLLControl MTI instances**

InstanceName	ComponentName
<code>PLLControl.clkdiv</code>	<code>ClockDivider</code>

## Ports for PLLControl

**Table 3-174: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	-
clk_out	<a href="#">ClockSignal</a>	Master	-
lock	<a href="#">Signal</a>	Master	-
rate	<a href="#">ClockRateControl</a>	Slave	-
unlock	<a href="#">Signal</a>	Master	-

## Parameters for PLLControl

### **clkdiv.div**

#### Type

int

#### Default value

0x1

#### Description

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

### **clkdiv.mul**

#### Type

int

#### Default value

0x1

#### Description

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

## 3.4.11 ScalableClockControl

Clock control allows input selection, rate control and gating. This model is written in C++.

### Iris and MTI instances for ScalableClockControl

This model has the following Iris instances:

**Table 3-175: ScalableClockControl Iris instances**

InstanceName	ComponentName
ScalableClockControl	ScalableClockControl
ScalableClockControl.clkDiv1	ClockDivider
ScalableClockControl.clkDiv10	ClockDivider
ScalableClockControl.clkDiv2	ClockDivider

InstanceName	ComponentName
ScalableClockControl.clkDiv3	ClockDivider
ScalableClockControl.clkDiv4	ClockDivider
ScalableClockControl.clkDiv5	ClockDivider
ScalableClockControl.clkDiv6	ClockDivider
ScalableClockControl.clkDiv7	ClockDivider
ScalableClockControl.clkDiv8	ClockDivider
ScalableClockControl.clkDiv9	ClockDivider
ScalableClockControl.clkGate	ClockGate
ScalableClockControl.clkGate.divider	ClockDivider
ScalableClockControl.clkSelect	ClockSelector
ScalableClockControl.clkSelect.clkdiv0	ClockDivider
ScalableClockControl.clkSelect.clkdiv1	ClockDivider
ScalableClockControl.clkSelect.clkdiv10	ClockDivider
ScalableClockControl.clkSelect.clkdiv2	ClockDivider
ScalableClockControl.clkSelect.clkdiv3	ClockDivider
ScalableClockControl.clkSelect.clkdiv4	ClockDivider
ScalableClockControl.clkSelect.clkdiv5	ClockDivider
ScalableClockControl.clkSelect.clkdiv6	ClockDivider
ScalableClockControl.clkSelect.clkdiv7	ClockDivider
ScalableClockControl.clkSelect.clkdiv8	ClockDivider
ScalableClockControl.clkSelect.clkdiv9	ClockDivider
ScalableClockControl.clkSelect.clkdivider	ClockDivider

This model has the following MTI trace components:

**Table 3-176: ScalableClockControl MTI instances**

InstanceName	ComponentName
ScalableClockControl.clkDiv1	ClockDivider
ScalableClockControl.clkDiv10	ClockDivider
ScalableClockControl.clkDiv2	ClockDivider
ScalableClockControl.clkDiv3	ClockDivider
ScalableClockControl.clkDiv4	ClockDivider
ScalableClockControl.clkDiv5	ClockDivider
ScalableClockControl.clkDiv6	ClockDivider
ScalableClockControl.clkDiv7	ClockDivider
ScalableClockControl.clkDiv8	ClockDivider
ScalableClockControl.clkDiv9	ClockDivider
ScalableClockControl.clkGate.divider	ClockDivider
ScalableClockControl.clkSelect.clkdiv0	ClockDivider
ScalableClockControl.clkSelect.clkdiv1	ClockDivider

InstanceName	ComponentName
ScalableClockControl.clkSelect.clkdiv10	ClockDivider
ScalableClockControl.clkSelect.clkdiv2	ClockDivider
ScalableClockControl.clkSelect.clkdiv3	ClockDivider
ScalableClockControl.clkSelect.clkdiv4	ClockDivider
ScalableClockControl.clkSelect.clkdiv5	ClockDivider
ScalableClockControl.clkSelect.clkdiv6	ClockDivider
ScalableClockControl.clkSelect.clkdiv7	ClockDivider
ScalableClockControl.clkSelect.clkdiv8	ClockDivider
ScalableClockControl.clkSelect.clkdiv9	ClockDivider
ScalableClockControl.clkSelect.clkdivider	ClockDivider

## Ports for ScalableClockControl

**Table 3-177: Ports**

Name	Protocol	Type	Description
clk_out	ClockSignal	Master	-
clkEnable	Signal	Slave	-
clkSel	Value	Slave	-
clock_in[10]	ClockSignal	Slave	-
clock_rate[10]	ClockRateControl	Slave	-
halt	Signal	Master	-
refClk_in	ClockSignal	Slave	-

## Parameters for ScalableClockControl

### **clkDiv1.div**

#### Type

int

#### Default value

0x1

#### Description

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

### **clkDiv1.mul**

#### Type

int

#### Default value

0x1

#### Description

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv10.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv10.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv2.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv2.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv3.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.



**clkDiv3.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv4.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv4.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv5.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv5.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv6.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv6.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv7.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv7.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv8.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv8.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv9.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv9.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkGate.diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

**clkGate.divider.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkGate.divider.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv0.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv0.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv1.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv1.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv10.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv10.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv2.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv2.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv3.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv3.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv4.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv4.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv5.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv5.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv6.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv6.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv7.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv7.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv8.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv8.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv9.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv9.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdivider.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdivider.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.



**clkSelect.diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

## 3.4.12 SwitchedClockControl

Clock control allows input selection, rate control and gating. This model is written in C++.

### Iris and MTI instances for SwitchedClockControl

This model has the following Iris instances:

**Table 3-178: SwitchedClockControl Iris instances**

InstanceName	ComponentName
SwitchedClockControl	SwitchedClockControl
SwitchedClockControl.clkDiv1	ClockDivider
SwitchedClockControl.clkDiv2	ClockDivider
SwitchedClockControl.clkGate	ClockGate
SwitchedClockControl.clkGate.divider	ClockDivider
SwitchedClockControl.clkSelect	ClockSelector
SwitchedClockControl.clkSelect.clkdiv0	ClockDivider
SwitchedClockControl.clkSelect.clkdiv1	ClockDivider
SwitchedClockControl.clkSelect.clkdiv10	ClockDivider
SwitchedClockControl.clkSelect.clkdiv2	ClockDivider
SwitchedClockControl.clkSelect.clkdiv3	ClockDivider
SwitchedClockControl.clkSelect.clkdiv4	ClockDivider
SwitchedClockControl.clkSelect.clkdiv5	ClockDivider
SwitchedClockControl.clkSelect.clkdiv6	ClockDivider
SwitchedClockControl.clkSelect.clkdiv7	ClockDivider
SwitchedClockControl.clkSelect.clkdiv8	ClockDivider
SwitchedClockControl.clkSelect.clkdiv9	ClockDivider
SwitchedClockControl.clkSelect.clkdivider	ClockDivider

This model has the following MTI trace components:

**Table 3-179: SwitchedClockControl MTI instances**

InstanceName	ComponentName
SwitchedClockControl.clkDiv1	ClockDivider

InstanceName	ComponentName
SwitchedClockControl.clkDiv2	ClockDivider
SwitchedClockControl.clkGate.divider	ClockDivider
SwitchedClockControl.clkSelect.clkdiv0	ClockDivider
SwitchedClockControl.clkSelect.clkdiv1	ClockDivider
SwitchedClockControl.clkSelect.clkdiv10	ClockDivider
SwitchedClockControl.clkSelect.clkdiv2	ClockDivider
SwitchedClockControl.clkSelect.clkdiv3	ClockDivider
SwitchedClockControl.clkSelect.clkdiv4	ClockDivider
SwitchedClockControl.clkSelect.clkdiv5	ClockDivider
SwitchedClockControl.clkSelect.clkdiv6	ClockDivider
SwitchedClockControl.clkSelect.clkdiv7	ClockDivider
SwitchedClockControl.clkSelect.clkdiv8	ClockDivider
SwitchedClockControl.clkSelect.clkdiv9	ClockDivider
SwitchedClockControl.clkSelect.clkdivider	ClockDivider

## Ports for SwitchedClockControl

**Table 3-180: Ports**

Name	Protocol	Type	Description
clk_out	ClockSignal	Master	-
clkDivExt	ClockRateControl	Slave	-
clkDivSys	ClockRateControl	Slave	-
clkEnable	Signal	Slave	-
clkSel	Value	Slave	-
halt	Signal	Master	-
refClk_in	ClockSignal	Slave	-
sysClk_in	ClockSignal	Slave	-
xClk_in	ClockSignal	Slave	-

## Parameters for SwitchedClockControl

### clkDiv1.div

#### Type

int

#### Default value

0x1

#### Description

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv1.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv2.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkDiv2.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkGate.diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

**clkGate.divider.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkGate.divider.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv0.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv0.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv1.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv1.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv10.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv10.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv2.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv2.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv3.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv3.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv4.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv4.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv5.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv5.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv6.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv6.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv7.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv7.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv8.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv8.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv9.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdiv9.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdivider.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkSelect.clkdivider.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.



**clkSelect.diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

## 3.5 Core components

This section describes the Core components.

Code Translation (CT) components translate instructions on the fly and cache the translations to enable fast execution of code, but sacrificing timing accuracy. They also use efficient PV bus models to enable fast access to memory and devices.

CT components differ in some ways from the modeled IP:

- They do not model cycle timing. In aggregate, all instructions execute in one processor master clock cycle, except for Wait For Interrupt.
- Write buffers are not modeled on all processors.
- Most aspects of TLB behavior are implemented in the models. In Arm®v7 models and later, the TLB memory attribute settings are used when stateful cache is enabled.
- No device-accurate MicroTLB is implemented.
- Device-accurate modeling of multiple TLBs is off by default.
- A single memory access port is implemented. The port combines accesses for instruction, data, DMA, and peripherals. Configuration of the peripheral port memory map register is ignored.
- All memory accesses are atomic and are performed in Programmer's View order. Unaligned accesses are always performed as byte transfers.
- Some instruction sequences are executed atomically so that system time does not advance during their execution. This difference in behavior can affect sequential accesses of device registers where devices are expecting time to move on between each access.
- Interrupts are not taken at every instruction boundary.
- Integration and test registers are not implemented.
- Models do not support running Software Test Libraries (STLs).
- Not all CP14 debug registers are implemented on all processors.
- Breakpoint types that the models support directly are:
  - Single address unconditional instruction breakpoints.
  - Single address unconditional data breakpoints.
  - Unconditional instruction address range breakpoints.

- Pseudoregisters in the debugger support processor exception breakpoints. Setting an exception register to a nonzero value stops execution on entry to the associated exception vector.
- Cluster models do not simulate all cores at the same time. They execute a number of instructions on each core in turn. There can be a bias in the order in which cores run after a restart (for example, core 0 always runs first), so the simulation might hit breakpoints on the favored core more often.
- Performance counters are not implemented on all models.
- Some models implement caches, although all processor models implement cache control registers.
- ECC and parity schemes are hardware-specific so are not modeled.
- Models use a simplified view of the external buses.
- Clusters based on DSU-110 and later support OFF, ON, AND DBG\_RECOV modes.

### 3.5.1 AEMv8RMPCT

AEMv8RMPCT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-181: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About AEMv8RMPCT

AEMv8RMPCT allows you to target AArch32 or AArch64, RAS, VFP, EL2, and other Arm®v8-R features.

This table lists the major differences between AEMv8RMPCT and Arm Cortex®-R82 Fast Models:

**Table 3-182: Major differences between AEMv8RMPCT and Arm Cortex-R82 Fast Models**

Feature	AEMv8R	Cortex-R82
VMSA_supported	Configurable	Not available
has_aarch64	Configurable	Not available
has_pl2	Configurable	Always true
has_pmu	Configurable	Always true
has_ras	Configurable	Always true
PA_SIZE	Configurable	Always 40
Armv8.5 feature-specific parameters	Configurable	Not available
stage1_tlb_size	Configurable	Not available
stage12_tlb_size	Configurable	Not available

Feature	AEMv8R	Cortex-R82
has_writebuffer	Configurable	Not available
vfp-present	Configurable	Always true
def_mem_map	Configurable	Fixed, according to the specification
IMPDEF registers, for example all IMP_* registers in the R82 specification.	Not available	Supported

This model supports the Arm®v8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for AEMv8RMPCT

This model has the following Iris instances:

**Table 3-183: AEMv8RMPCT Iris instances**

InstanceName	ComponentName
AEMv8RMPCT	Cluster_ARMAEMv8-R_MP
AEMv8RMPCT.AMU	PVBusLogger
AEMv8RMPCT.AMU.mapper	PVBusMapper
AEMv8RMPCT.DAP	PVBusLogger
AEMv8RMPCT.DAP.mapper	PVBusMapper
AEMv8RMPCT.MMAP	PVBusLogger
AEMv8RMPCT.MMAP.mapper	PVBusMapper
AEMv8RMPCT.RAS	PVBusLogger
AEMv8RMPCT.RAS.mapper	PVBusMapper
AEMv8RMPCT.acp_mapper	PVBusMapper
AEMv8RMPCT.cpu0	ARMAEMv8-R_MP
AEMv8RMPCT.cpu0.UTLB	TLB
AEMv8RMPCT.cpu0.debug_rom	debug_rom
AEMv8RMPCT.cpu0.dtlb	TLB
AEMv8RMPCT.cpu0.l1dcache	PVCache
AEMv8RMPCT.cpu0.l1dcache.upstream[0]	PVBusSlave
AEMv8RMPCT.cpu0.l1icache	PVCache
AEMv8RMPCT.cpu0.l1icache.upstream[0]	PVBusSlave
AEMv8RMPCT.ext_bus	PVBusLogger
AEMv8RMPCT.ext_bus.mapper	PVBusMapper
AEMv8RMPCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
AEMv8RMPCT.global_debug_rom	debug_rom
AEMv8RMPCT.l2_cache	PVCache
AEMv8RMPCT.l2_cache.upstream[0]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[10]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[11]	PVBusSlave

InstanceName	ComponentName
AEMv8RMPCT.l2_cache.upstream[12]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[13]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[14]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[15]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[16]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[1]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[2]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[3]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[4]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[5]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[6]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[7]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[8]	PVBusSlave
AEMv8RMPCT.l2_cache.upstream[9]	PVBusSlave
AEMv8RMPCT.l2_flusher	AsyncCacheFlushUnit
AEMv8RMPCT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-184: AEMv8RMPCT MTI instances**

InstanceName	ComponentName
AEMv8RMPCT.AMU	PVBusLogger
AEMv8RMPCT.AMU.mapper	PVBusMapper
AEMv8RMPCT.DAP	PVBusLogger
AEMv8RMPCT.DAP.mapper	PVBusMapper
AEMv8RMPCT.MMAP	PVBusLogger
AEMv8RMPCT.MMAP.mapper	PVBusMapper
AEMv8RMPCT.RAS	PVBusLogger
AEMv8RMPCT.RAS.mapper	PVBusMapper
AEMv8RMPCT.acp_mapper	PVBusMapper
AEMv8RMPCT.cpu0	ARM_AEMv8-R_MP
AEMv8RMPCT.cpu0.UTLB	TLB
AEMv8RMPCT.cpu0.l1dcache	PVCache
AEMv8RMPCT.cpu0.l1dcache.upstream[0]	PVBusSlave
AEMv8RMPCT.cpu0.l1icache	PVCache
AEMv8RMPCT.cpu0.l1icache.upstream[0]	PVBusSlave
AEMv8RMPCT.ext_bus	PVBusLogger
AEMv8RMPCT.ext_bus.mapper	PVBusMapper
AEMv8RMPCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
AEMv8RMPCT.l2_cache	PVCache

InstanceName	ComponentName
AEMv8RMPCT.12_cache.upstream[0]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[10]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[11]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[12]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[13]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[14]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[15]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[16]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[1]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[2]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[3]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[4]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[5]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[6]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[7]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[8]	PVBusSlave
AEMv8RMPCT.12_cache.upstream[9]	PVBusSlave
AEMv8RMPCT.12_flusher	AsyncCacheFlushUnit

## Ports for AEMv8RMPCT

**Table 3-185: Ports**

Name	Protocol	Type	Description
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	Signal	Slave	Enable broadcasting of Inner Shareable transactions.
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
cache_validation_control	Value	Slave	This signal provides default exception handling state.
cfgend[4]	Signal	Slave	This signal if for EE bit initialisation
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC
CNTHPSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC

Name	Protocol	Type	Description
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module
CNTVIRQ[4]	Signal	Master	Timer signals to SOC
commirq[4]	Signal	Master	Interrupt signal from debug communication channel.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	-
cti0extin[4]	Signal	Slave	-
cti0extout[4]	Signal	Master	-
cti1extin[4]	Signal	Slave	-
cti1extout[4]	Signal	Master	-
cti2extin[4]	Signal	Slave	-
cti2extout[4]	Signal	Master	-
cti3extin[4]	Signal	Slave	-
cti3extout[4]	Signal	Master	-
ctidbgirq[4]	Signal	Master	-
dbgen[4]	Signal	Slave	-
dbgnopwrdown[4]	Signal	Master	These signals relate to core power down.
dbgpwrdownack[4]	Signal	Master	Debug power down acknowledge.
dbgpwrdownreq[4]	Signal	Slave	Debug power down request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
ext_slave_s	PVBus	Slave	External Slave port. Equivalent to AXIS port.
external_trace_reset[4]	Signal	Slave	ETMv4 External Trace Reset signal.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
flash_m[4]	PVBus	Master	Flash Port
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
hiden[4]	Signal	Slave	External debug interface.
hniden[4]	Signal	Slave	External debug interface.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
irqs[224]	Signal	Slave	These signals drive the CPU's interrupt controller interrupt lines.

Name	Protocol	Type	Description
l2reset	Signal	Slave	This signal resets timer and interrupt controller and l2cache
llpp_m[4]	PVBus	Master	LLPP (Low-Latency Peripheral Port).
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg[4]	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	-
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
rvbar[4]	Value_64	Slave	Reset vector base address.
sei[4]	Signal	Slave	Per core System Error physical pins
smpnamp[4]	Signal	Master	This signals AMP or SMP mode for each core
spiden[4]	Signal	Slave	Secure invasive debug enable.
spniden[4]	Signal	Slave	Secure non-invasive debug enable.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state
teinit[4]	Signal	Slave	This signal provides default exception handling state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trace_unit_reset[4]	Signal	Slave	ETMv4 Trace Unit Reset signal.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for AEMv8RMPCT

### **ADFSR-AIFSR-implemented**

**Type**

bool

**Default value**

0x0

**Description**

ADFSR and AIFSR are implemented.

### **AIDR**

**Type**

int

**Default value**

0x0

**Description**

Value of AIDR\_EL1 register.

### **AMIIDR**

**Type**

int

**Default value**

0x43b

**Description**

Value of AMU Implementation Identification Register.

### **AMPIDR**

**Type**

int

**Default value**

0x4000bb000

**Description**

Value of AMU Peripheral Identification Register.

### **BPIMVA\_causes\_translation\_lookup**

**Type**

bool

**Default value**

0x0



**Description**

Do a translation when BPIMVA instruction is executed (which may cause a translation fault).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation to NC/DEV memory. The broadcastatomic signal will override this value if used.

**BROADCASTATOMICL****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation to LLRAM memory. The broadcastatomicl signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTINNER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**CCSIDR-L1D\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L1I\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L2\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L3\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, allow L3 selection in CSSELR and present this value in CCSIDR (this is cosmetic and does not affect cache behaviour).

**CFGTFPEN\_pin\_reset****Type**

bool

**Default value**

0x0

**Description**

CFGTFPEN Configuration pin at reset for bitfield IMP\_MEMPROTCTLR\_EL1.TFPEN.

**CHI****Type**

bool

**Default value**

0x0

**Description**

Selects the type of protocol the Main Manager(MM) interface implements. 0, MM port configured as AXI. 1, MM port configured as CHI.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**CTIPIDR****Type**

int

**Default value**

0x0

**Description**

If non-zero, override the CTI Peripheral Identification Register.

**CTR-L1Ip-override****Type**

int

**Default value**

0x0

**Description**

If non-zero, override the L1Ip bits in CTR/CTR\_ELO system register. This does not change the behaviour of the cache, only what is present in the CTR register.

**DBGBCR\_BT\_applies\_RES0\_before\_valid\_check****Type**

bool

**Default value**

0x1

**Description**

If true, RESO behaviour is applied to DBGBCR(\_EL1).BT before checking for reserved values for this field.

**DBGPIDR****Type**

int

**Default value**

0x0

**Description**

If non-zero, override the Debug Peripheral Identification Register.

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x0

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**ERRIIDR****Type**

int

**Default value**

0xd800143b

**Description**

Value of RAS Implementation Identification Register.

**ERRPIDR****Type**

int

**Default value**

0x4100bbd80

**Description**

Value of RAS Peripheral Identification Register.

**ERXMISCO\_mask****Type**

int

**Default value**

0x0

**Description**

Write Mask for ERXMISCO RAS Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**MIDR****Type**

int

**Default value**

0x410fd0f0

**Description**

Value of MIDR\_EL1 register.

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in cluster.

**PA\_SIZE****Type**

int

**Default value**

0x28

**Description**

Physical address range supported (FEAT\_LPA).

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**PMCEID0****Type**

int

**Default value**

0xffffffff

**Description**

Performance Monitor Common Event ID Reg 0 value - 64 bit.

**PMCEID1****Type**

int

**Default value**

0xffffffff

**Description**

Performance Monitor Common Event ID Reg 1 value - 64 bit.

**PMSIDR.ArchInst****Type**

bool

**Default value**

0x1

**Description**

Defines whether architecture instruction sampling is implemented or not, if not only micro op sampling is implemented. Model only supports architecture instruction sampling, but allows ID register field to be configured.

**PMSIDR.CRR****Type**

bool

**Default value**

0x0

**Description**

Defines whether call return branch records (FEAT\_SPE\_CRR) is implemented or not.

**PMSIDR.LDS****Type**

bool

**Default value**

0x0

**Description**

Defines whether data source for sampled load instruction is implemented or not. Model does not implement loaded data source, but allows ID register field to be configured.

**PMUPIDR****Type**

int

**Default value**

0x0

**Description**

If non-zero, override the PMU Peripheral Identification Register.

**VMSA\_supported****Type**

bool

**Default value**

0x1

**Description**

VMSA is supported at EL1.



**abort\_execution\_from\_device\_memory****Type**

bool

**Default value**

0x0

**Description**

Execution from device memory generates a prefetch abort.

**advsimd\_overread****Type**

bool

**Default value**

0x0

**Description**

AdvSIMD element load operations access all bytes of a 16-byte aligned window, even in Device memory.

**align\_pc\_on\_branch\_to\_unaligned\_pc\_aarch32****Type**

bool

**Default value**

0x0

**Description**

Force PC align for branches to an unaligned PC counter in A32 state.

**align\_pc\_on\_debug\_exit\_to\_aarch32****Type**

bool

**Default value**

0x0

**Description**

Exit to AARCH32 state from debug state forces pc bit0 to 0.

**align\_pc\_on\_illegal\_exception\_return\_to\_aarch32****Type**

bool

**Default value**

0x1

**Description**

Align PC when performing an illegal exception return from AArch64 to AArch32.

**amu\_aux\_type\_fixed****Type**

string

**Default value**

""

**Description**

Lists which AMU auxiliary registers that are fixed and to which event type. The JSON schema is: {fixed\_aux\_reg:evt\_type, ...}. For example {"0":0x300} would make auxiliary register 0 fixed to event type 0x300.

**amu\_mmap\_address****Type**

string

**Default value**

""

**Description**

AMU base address for each core on system bus. 0 means the AMU is not mapped, otherwise the address must be 4KB aligned. JSON schema for the parameter value is: {"format":"all\_addrs\_are\_absolute\_wrt\_systembus", "cores": [{"amu":0x0}, {"amu":0x0}, {"amu":0x0}, {"amu":0x0}]}.

**amu\_num\_auxiliary\_counters****Type**

int

**Default value**

0x0

**Description**

Number of AMU auxiliary counters implemented.

**apshr\_read\_restrict****Type**

bool

**Default value**

0x0

**Description**

At EL0, unknown bits of APSR are RAZ.

**atomic\_memtype\_fault\_prio\_less\_than\_gpc\_fault****Type**

bool

**Default value**

0x0

**Description**

If true, unsupported atomic/exclusive memtype faults are lower priority than GPC faults.

**atomic\_memtype\_fault\_priority****Type**

int

**Default value**

0x0

**Description**

This parameter describes the priority of unsupported atomic/exclusive memtype fault w.r.t alignment and permission fault. 0, BEFORE\_ALIGN\_MEM\_FAULT. 1, AFTER\_ALIGN\_BEFORE\_PERM\_FAULT. 2, AFTER\_PERM\_FAULT.

**auxilliary\_feature\_register0****Type**

int

**Default value**

0x0

**Description**

Value of AFR0 ID register.

**branch-predictor-clear-policy****Type**

int

**Default value**

0x2

**Description**

Set branch prediction policy as defined for MMFR1[31:28]. This does not change the behaviour of the branch predictor, only what is reported in MMFR1.BPred.

**branch-predictor-supported-ops****Type**

int

**Default value**

0x1

**Description**

Set branch prediction policy as defined for MMFR3[11:8]. This does not change the behaviour of the branch predictor, only what is reported in MMFR3.BPMaint.

**bus\_protection\_enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable TCM, L1Cache bus protection after reset.

**cache-log2linelen****Type**

int

**Default value**

0x6

**Description**

Log2 of the cache line length in bytes.

**cache\_maintenance\_hits\_watchpoints****Type**

bool

**Default value**

0x0

**Description**

DCIMVA operations executed in AArch32 modes hit watchpoints.

**changing\_block\_size\_without\_bbm\_support****Type**

int

**Default value**

0x0

**Description**

Level of support for changing block size without break-before-make (FEAT\_BBM).

**check\_memory\_attributes****Type**

bool

**Default value**

0x0

**Description**

Detect and report TLB use of conflicting memory attributes for views of the same physical address.

**`clean_invalidate_cache_on_warm_reset`****Type**

bool

**Default value**

0x0

**Description**

Clean and invalidate caches on warm reset.

**`clear_reg_top_eret`****Type**

int

**Default value**

0x1

**Description**

Behaviour of the upper 32-bits of the Xn registers when changing between AArch32 state and AArch64 state. 0, upper 32-bits preserved for all registers. 1, upper 32-bits set to 0 for all accessible registers. 2, upper 32-bits set to 0 for a random selection of accessible registers. 3, upper-32-bits set to 0 for registers touched in AArch32.

**`clear_reg_top_set`****Type**

bool

**Default value**

0x1

**Description**

Whether to clear upper 32-bits of the Xn register when corresponding AArch32 register is set via CADI/Iris.

**`cluster_utid`****Type**

int

**Default value**

0x0

**Description**

Unique cluster transaction identifier for interconnect protection.

**configure\_pmu\_events\_with\_json****Type**

string

**Default value**

""

**Description**

"Configure v8.6 and newer PMU events. Note : This param has high priority and overrides the setting of "has\_\*\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["EVENT\_NAME\_1","EVENT\_NAME\_2"]}".

**configure\_v8\_6\_pmu\_events\_with\_json****Type**

string

**Default value**

""

**Description**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.6 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_6\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["BR\_INDNR\_RETIRE", "BR\_IND\_RETIRE", "BR\_RETURN\_SKIP\_RETIRE", "BR\_RETURN\_ANY\_RETIRE", "BR\_INDNR\_SKIP\_RETIRE", "BR\_INDNR\_TAKEN\_RETIRE", "BR\_IND\_SKIP\_RETIRE", "BR\_IND\_TAKEN\_RETIRE", "BR\_IMMED\_SKIP\_RETIRE", "BR\_IMMED\_TAKEN\_RETIRE", "BR\_SKIP\_RETIRE"]}".

**configure\_v8\_8\_pmu\_events\_with\_json****Type**

string

**Default value**

""

**Description**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.8 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_8\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["BR\_HINT\_COND\_RETIRE", "BR\_COND\_TAKEN\_RETIRE", "BR\_UNCOND\_RETIRE", "BR\_COND\_RETIRE", "BRNL\_TAKEN\_RETIRE", "BRNL\_IND\_TAKEN\_RETIRE", "BRNL\_INDNR\_TAKEN\_RETIRE", "BRNL\_IMMED\_TAKEN\_RETIRE", "BL\_TAKEN\_RETIRE", "BL\_IND\_TAKEN\_RETIRE", "BL\_IMMED\_TAKEN\_RETIRE"]}".

**configure\_v8\_9\_pmu\_events\_with\_json****Type**

string

**Default value**

""

**Description**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.9 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_9\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["ASE\_SVE\_RETIRE", "ASE\_RETIRE", "VFP\_RETIRE", "SVE\_RETIRE", "CRYPTO\_RETIRE", "SIMD\_INST\_RETIRE", "ASE\_INST\_RETIRE", "SVE\_INST\_RETIRE", "ASE\_SVE\_INST\_RETIRE", "LD\_ANY\_RETIRE", "ST\_ANY\_RETIRE", "LDST\_ANY\_RETIRE", "DP\_RETIRE"]}"

**core\_cache\_protection****Type**

int

**Default value**

0xffffffffffffffff

**Description**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

**cpacr\_trcdis\_behaviour****Type**

int

**Default value**

0x2

**Description**

Behaviour of CPACR.TRCDIS/NSACR.NSTRCDIS when there is no CP14 ETM interface. 0, RAZ/WI. 2, implemented.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.CP15SDISABLE2****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers (FEAT\_CP15SDISABLE2).

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.



**cpuX.DCZID-log2-block-size****Type**

int

**Default value**

0x8

**Description**

Log2 of the block size cleared by DC ZVA instruction (as read from DCZID\_EL0).

**cpuX.DCZVA\_single\_write****Type**

bool

**Default value**

0x0

**Description**

Execute the DCZVA as a single write.

**cpuX.MPIDR-override****Type**

int

**Default value**

0x0

**Description**

Override of MPIDR value. If nonzero will override the MT, cluster and CPU ID bits in MPIDR.

**cpuX.RVBAR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.RVBAR32****Type**

int

**Default value**

0x0

**Description**

Reset vector address in AARCH32 when VINITHI is not set and ignore\_rvbar\_in\_aarch32 is set.

**cpuX.SMPnAMP****Type**

bool

**Default value**

0x1

**Description**

Enable broadcast messages necessary for correct SMP operation at reset.

**cpuX.TEINIT****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTL.R.V.

**cpuX.aarch32\_reset\_from\_impdef\_addr****Type**

bool

**Default value**

0x1

**Description**

If PE resets into AArch32, Whether execution starts from IMPDEF address or hi/low vector.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has been built with NEON support.

**`cpuX.clock_divider`****Type**

int

**Default value**

0x1

**Description**

Clock divider ratio for asymmetric MP clocking.

**`cpuX.clock_multiplier`****Type**

int

**Default value**

0x1

**Description**

Clock divider ratio for asymmetric MP clocking.

**`cpuX.crypto_aes`****Type**

int

**Default value**

0x2

**Description**

AES instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 2, AES and PMULL instructions implemented (FEAT\_AES, FEAT\_PMULL).

**`cpuX.crypto_sha1`****Type**

int

**Default value**

0x1

**Description**

SHA-1 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA1 instructions implemented (FEAT\_SHA1).

**cpuX.crypto\_sha256****Type**

int

**Default value**

0x1

**Description**

SHA-256 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA256 instructions implemented (FEAT\_SHA256).

**cpuX.cti-intack\_mask****Type**

int

**Default value**

0x1

**Description**

Set bits represent that the corresponding trigger requires software acknowledge via CTIINTACK.

**cpuX.cti-number\_of\_claim\_bits****Type**

int

**Default value**

0x0

**Description**

Number of implemented bits in CTICLAIMSET.

**cpuX.cti-number\_of\_triggers****Type**

int

**Default value**

0x8

**Description**

Number of cti event triggers (default: 8, valid values: {3, 8-32}).

**cpuX.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**cpuX.enable\_crc32****Type**

int

**Default value**

0x0

**Description**

CRC32 instructions supported. 0, not implemented. 1, CRC32 instructions implemented (FEAT\_CRC32).

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.etm-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has ETM support.

**cpuX.flash.enable****Type**

bool

**Default value**

0x0

**Description**

Enable flash by default after reset.

**cpuX.force-fpsid****Type**

bool

**Default value**

0x0

**Description**

Override the FPSID value.

**cpuX.force-fpsid-value****Type**

int

**Default value**

0x0

**Description**

Value to override the FPSID value to.

**cpuX.has\_hcptr\_tase****Type**

bool

**Default value**

0x1

**Description**

If false, HCPTR.TASE is RES0.

**cpuX.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**cpuX.llpp.base****Type**

int

**Default value**

0x0

**Description**

Sets the base address of Low Latency Peripheral Port.

**cpuX.llpp.size****Type**

int

**Default value**

0x1000

**Description**

Sets the size of LLPP(in bytes).

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.number-of-breakpoints****Type**

int

**Default value**

0x10

**Description**

Number of breakpoints.

**cpuX.number-of-context-breakpoints****Type**

int

**Default value**

0x10

**Description**

Number of breakpoints that are context aware.

**cpuX.number-of-watchpoints****Type**

int

**Default value**

0x10

**Description**

Number of watchpoints.

**cpuX.operation\_bandwidth****Type**

int

**Default value**

0x1

**Description**

Operation width for ARMv8.4 PMU extension.

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.



**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.semihosting-stderr\_istty****Type**

bool

**Default value**

0x1

**Description**

Result for semihost istty call when argument is stderr.

**cpuX.semihosting-stdin\_istty****Type**

bool

**Default value**

0x1

**Description**

Result for semihost istty call when argument is stdin.

**cpuX.semihosting-stdout\_istty****Type**

bool

**Default value**

0x1

**Description**

Result for semihost istty call when argument is stdout.

**cpuX.semihosting-use\_stderr****Type**

bool

**Default value**

0x0

**Description**

Send stderr from the simulated process to host stderr.

**cpuX.tcm-present****Type**

bool

**Default value**

0x0

**Description**

Disables the TCMs.

**cpuX.tcm-supports-exclusive****Type**

bool

**Default value**

0x0

**Description**

Whether TCM supports exclusive access.

**cpuX.tcm.a.base****Type**

int

**Default value**

0x0

**Description**

Sets the base address of the ATCM. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

**cpuX.tcm.a.enable****Type**

bool

**Default value**

0x0

**Description**

Enable ATCM by default after reset. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

**cpuX.tcm.a.size****Type**

int

**Default value**

0x4000

**Description**

Sets the size of the ATCM(in bytes). The size should be aligned with 4KB. The lower 12 bits will be masked out.

**cpuX.tcm.a.stretch\_clk****Type**

bool

**Default value**

0x0

**Description**

Whether ATCM clock stretched to occupy full cycle.

**cpuX.tcm.a.wait****Type**

int

**Default value**

0x0

**Description**

ATCM accesses wait states.

**cpuX.tcm.b.base****Type**

int

**Default value**

0x0

**Description**

Sets the base address of the BTCM. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

**cpuX.tcm.b.enable****Type**

bool

**Default value**

0x0

**Description**

Enable BTCM by default after reset. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

**cpuX.tcm.b.size****Type**

int

**Default value**

0x4000

**Description**

Sets the size of the BTCM(in bytes). The size should be aligned with 4KB. The lower 12 bits will be masked out.

**cpuX.tcm.b.stretch\_clk****Type**

bool

**Default value**

0x0

**Description**

Whether BTCM clock stretched to occupy full cycle.

**cpuX.tcm.b.wait****Type**

int

**Default value**

0x0

**Description**

BTCM accesses wait states.

**cpuX.tcm.c.base****Type**

int

**Default value**

0x0

**Description**

Sets the base address of the CTCM. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

**cpuX.tcm.c.enable****Type**

bool

**Default value**

0x0

**Description**

Enable CTCM by default after reset. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

**cpuX.tcm.c.size****Type**

int

**Default value**

0x2000

**Description**

Sets the size of the CTCM(in bytes). The size should be aligned with 4KB. The lower 12 bits will be masked out.

**cpuX.tcm.c.stretch\_clk****Type**

bool

**Default value**

0x0

**Description**

Whether CTCM clock stretched to occupy full cycle.

**cpuX.tcm.c.wait****Type**

int

**Default value**

0x0

**Description**

CTCM accesses wait states.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.unpredictable\_WPMASKANDBAS****Type**

int

**Default value**

0x1

**Description**

Constrained unpredictable handling of watchpoints when mask and BAS fields specified. 0, IGNOREMASK. 1, IGNOREBAS (default). 2, REPEATBAS8. 3, REPEATBAS.

**cpuX.vfp-dp-present****Type**

bool

**Default value**

0x1

**Description**

Whether double-precision floating point feature is implemented (FEAT\_F64MM).

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**cpuX.vfp-traps****Type**

bool

**Default value**

0x1

**Description**

Implement support for trapping floating-point exceptions.

**cpuX.vfp-traps-show-all****Type**

bool

**Default value**

0x0



**Description**

Report all trapped floating-point exceptions in the syndrome when a combination occurs.

**cpuX.wfet\_early\_or\_delayed\_timeout****Type**

int

**Default value**

0x0

**Description**

WFET early or delayed timeout beyond the threshold value of CNTVCT\_ELO in percentage.

**cpuX.wfit\_early\_or\_delayed\_timeout****Type**

int

**Default value**

0x0

**Description**

WFIT early or delayed timeout beyond the threshold value of CNTVCT\_ELO in percentage.

**dbg-bcr-reserved-behavior****Type**

int

**Default value**

0x1

**Description**

This is the behavior of the reserved values of the BT field in DBGBCR. Possible values are: - 0 = Disabled. - 1 = BT[2] is ignored. .

**dbg\_rom\_dap\_addr****Type**

int

**Default value**

0x0

**Description**

Debug ROM dap base address.

**dbgitr\_buffer\_size****Type**

int

**Default value**

0x0

**Description**

Number of instructions which can be buffered before EDSCR.ITE is cleared.

**dbgxvr\_ress\_is\_stateful****Type**

bool

**Default value**

0x0

**Description**

Whether DBGWVR/DBGBVR.RESS returns last written value. if set to false, RESS returns sign extended value.

**dc\_fault\_unaligned\_s1\_device\_s2\_fwb****Type**

bool

**Default value**

0x0

**Description**

Whether takes an Alignment Fault caused by the memory type on a DC {ZVA,GZVA,GVA} if the stage 1 memory type is any Device memory type.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x8

**Description**

L1 D-Cache read bus width in bytes used to calculate per-access timing annotations.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-ways****Type**

int

**Default value**

0x2

**Description**

L1 D-Cache number of ways (sets are implicit from size).

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x8

**Description**

L1 D-Cache write bus width in bytes used to calculate per-access timing annotations.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**dcimva\_requires\_s2\_write\_permissions****Type**

bool

**Default value**

0x0

**Description**

Data-cache invalidate by MVA operations require stage 2 write permission (virtualised AArch32 guest).

**debug\_auth\_signals\_sampled\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Debug authentication signals can be configured as either sampled at reset only or at any time for External Root Debug.

**debug\_components\_dap\_address****Type**

string

**Default value**

""

**Description**

Debug components ROM,ED,CTI,PMU,TRACE and TRBU base address for each core on debug bus. The "rom" field in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. JSON schema for the parameter value is: {"format":"all\_addrs\_are\_absolute\_wrt\_debugbus", "cores": [{"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}]}.

**debug\_components\_mmap\_address****Type**

string

**Default value**

""

**Description**

Debug components ROM,ED,CTI,PMU,TRACE and TRBU base address for each core on system bus. The "rom" field in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. JSON schema for the parameter value is: {"format":"all\_addrs\_are\_absolute\_wrt\_systembus", "cores": [{"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}]}.

**debug\_entry\_is\_context\_sync****Type**

bool

**Default value**

0x0

**Description**

If true, Entry in debug state is Context sync. Exiting debug state is a context synchronising operation, but entering is not. However some cpu implementation can consider also the Entry in Debug state as a CSE.

**debug\_rom\_is\_class\_9****Type**

bool

**Default value**

0x0

**Description**

If true, present a debug ROM table as a class 9 device. Otherwise, use a class 1 ROM table.

**debug\_rom\_is\_flat****Type**

bool

**Default value**

0x0

**Description**

If true, present a debug ROM table recommended by ARMv8 Debug Architecture. Otherwise, use nested ROM tables.

**def\_mem\_map****Type**

string

**Default value**

{ "\*": "NORMAL INC ONC OSH" }

**Description**

Default memory map in a json format which is: { "start-end\_inclusive" : "attributes", "\*" : "attributes" } Where the \* represents the entire physical address range and must be provided and the attributes can be a combination of following. MemoryType - NORMAL, GRE, nGRE, nGnRE, nGnRnE Shareability - ISH, OSH, NSH InnerAttributes - IWB, IWT, INC OuterAttribute - OWB, OWT, ONC ExecuteNever - XN.

**def\_mem\_map\_file\_path****Type**

string

**Default value**

""

**Description**

Path of file describing default memory map in json format. When a valid path is provided, the below parameter 'def\_mem\_map' will be ignored.

**default\_inner\_shareable****Type**

bool

**Default value**

0x0

**Description**

shareability for default memory map regions which are shareable.

**delay\_serror****Type**

int

**Default value**

0x0

**Description**

Add a propagation delay of serror signal into the core.

**dic-spi\_count****Type**

int

**Default value**

0x40

**Description**

Number of shared peripheral interrupts implemented.

**disable\_impdef\_abort\_on\_ic\_maintenance****Type**

bool

**Default value**

0x0



**Description**

Disable IMP\_INTLATENCY\_EL2.MMDVM/LLRAMDVM controlling the abort on IC maintenance on MM port/LLRAM.

**`disable_sve_plugin`****Type**

bool

**Default value**

0x0

**Description**

If true, SVE will not be implemented in this processor even if the plugin is loaded (FEAT\_SVE).

**`disable_unknown_update_event_on_reset`****Type**

bool

**Default value**

0x0

**Description**

Disables SYSREG\_UPDATE event notification on reset for the registers whose bitfields are all reserved or resets to architecturally unknown value.

**`dsb_accumulate_threshold`****Type**

int

**Default value**

0x100

**Description**

Limit the maximum number of observable DSB side effects which can be queued (e.g. TLBI), after which DSB sync will be done automatically.

**`edpfr_ras_unknown_bits_read_as_0`****Type**

bool

**Default value**

0x0

**Description**

If true then UNKNOWN bits in RAS field in EDPFR are read as 0.

**e10\_can\_access\_imp\_def\_functionality****Type**

bool

**Default value**

0x0

**Description**

If not made UNDEF by imp\_def\_functionality\_behaviour, ELO can access IMPLEMENTATION DEFINED registers and system instructions.

**e13\_trap\_priority\_when\_secure\_debug\_disabled****Type**

bool

**Default value**

0x0

**Description**

Undef when secure debug is disabled (EDSCR.SDD == 1) && boolean IMPLEMENTATION\_DEFINED 'EL3 trap priority when SDD == 1'.

**enable-gicv5****Type**

bool

**Default value**

0x0

**Description**

if enable-gicv5 is set, then GICv5 is Supported.

**enable\_address\_contig\_check****Type**

bool

**Default value**

0x0

**Description**

Check the input address range for the table entries that have the contiguous hint bit set.

**enable\_debug\_auth\_signals\_config****Type**

int

**Default value**

0xf

**Description**

Debug Authentication Signals DBGEN, SPIDEN (and if RME is enabled RLPIDEN and RTPIDEN) are configurable (default) or not configurable, (hardwired to 1). This parameter is the integer representation of a bitmap to enable configuration of these signals, with: - BIT[0] = DBGEN - BIT[1] = SPIDEN - BIT[2] = RLPIDEN - BIT[3] = RTPIDEN .

**enable\_lock\_step****Type**

bool

**Default value**

0x0

**Description**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

**enable\_tlb\_contig\_check****Type**

bool

**Default value**

0x0

**Description**

Perform extra pagetable walks to check translation table entries that have the contiguous hint bit set.

**enhanced\_pac2\_level****Type**

int

**Default value**

0x0

**Description**

Implements Enhanced PAC2. 0: No EnhancedPAC2, 1: EnhancedPAC2 Only, 2: EnhancedPAC2 with FPAC, 3: EnhancedPAC2 with FPACCombined.

**error\_record\_feature\_register****Type**

string

**Default value**

""

**Description**

RAS feature register values. An array of JSON objects. The JSON schema for the array is: [{"ED":0x0, "IMPDEF\_3\_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0,

"UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other\_feature\_register\_values]. Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4. RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

### **error\_record\_feature\_register\_json\_file**

#### **Type**

string

#### **Default value**

""

#### **Description**

File path to the RAS feature register values as JSON. The file uses the same format as the error\_record\_feature\_register parameter value.

### **erxpfctl\_res0\_stateful\_mask**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

Mask for stateful bits for ERXPFGCTL which are RES0.

### **exception\_catch\_before\_software\_step**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

Exception catch priority for the exception trapping form of exception catch (Armv8.2 or later, or exception\_catch\_type=0). If true, the exception catch debug event has higher priority than software step and halting step.

### **exception\_catch\_type**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

Type of exception catch (ARMv8.0 - ARMv8.1 only). 0, exception trapping. 1, non-exception trapping, higher priority than step. 2, non-exception trapping, lower priority than step.

**exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Exclusive monitors in the cluster will be cleared by a atomic by the same master to the monitored address.

**exclusive\_monitor\_clear\_on\_store\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Exclusive monitors in the cluster will be cleared by a store by the same master to the monitored address.

**exclusive\_monitor\_clear\_on\_strex\_address\_mismatch****Type**

bool

**Default value**

0x1

**Description**

Exclusive monitors in the cluster will be cleared when a strex fails because the address does not match.

**exclusive\_monitor\_clear\_on\_strex\_success****Type**

bool

**Default value**

0x1

**Description**

Exclusive monitors in the cluster will be cleared when a strex succeeds.

**exercise\_stxr\_fail****Type**

int

**Default value**

0x0

**Description**

Controls the rejection of exclusive store instructions. 0: exclusive store instructions should behave as normal, 1: Reject a pseudo-random majority of exclusive store instructions, 2: Always fail exclusive store instructions.

**ext\_abort\_device\_GRE\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_GRE\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_device\_GRE\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-GRE read external aborts.

**ext\_abort\_device\_GRE\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of device-GRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_read\_is\_sync.

**ext\_abort\_device\_GRE\_read\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_GRE\_read\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_device\_GRE\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-GRE write external aborts.

**ext\_abort\_device\_GRE\_write\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of device-GRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_write\_is\_sync.

**ext\_abort\_device\_GRE\_write\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_GRE\_write\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_device\_nGRE\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_nGRE\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_device\_nGRE\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGRE read external aborts.



**ext\_abort\_device\_nGRE\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of device-nGRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_read\_is\_sync.

**ext\_abort\_device\_nGRE\_read\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_nGRE\_read\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_device\_nGRE\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGRE write external aborts.

**ext\_abort\_device\_nGRE\_write\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of device-nGRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_write\_is\_sync.

**ext\_abort\_device\_nGRE\_write\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_nGRE\_write\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_device\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_device\_read\_acquire\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device read with acquire external aborts.

**ext\_abort\_device\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_device\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGnRE write external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_device\_write\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_write\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_fill\_data****Type**

int

**Default value**

0xfdfdfdfcfcfdfdfd

**Description**

Returned data, if external aborts are asynchronous.

**ext\_abort\_normal\_cacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal write-back cacheable-read external aborts.

**ext\_abort\_normal\_cacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of normal write-back cacheable-read external aborts.

**ext\_abort\_normal\_cacheable\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_cacheable\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_normal\_cacheable\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal write-back cacheable write external aborts.

**ext\_abort\_normal\_cacheable\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of normal write-back cacheable write external aborts.

**ext\_abort\_normal\_cacheable\_write\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_cacheable\_write\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_normal\_noncacheable\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_noncacheable\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_normal\_noncacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal noncacheable-read external aborts.

**ext\_abort\_normal\_noncacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of normal noncacheable-read external aborts.

**ext\_abort\_normal\_noncacheable\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_noncacheable\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_normal\_noncacheable\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal noncacheable write external aborts.

**ext\_abort\_normal\_noncacheable\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of normal noncacheable write external aborts.

**ext\_abort\_normal\_noncacheable\_write\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].



**ext\_abort\_normal\_noncacheable\_write\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_normal\_wt\_cacheable\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_wt\_cacheable\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_normal\_wt\_cacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal write-through cacheable-read external aborts.

**ext\_abort\_normal\_wt\_cacheable\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of normal write-through read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_normal\_cacheable\_read\_is\_sync.

**ext\_abort\_normal\_wt\_cacheable\_read\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_wt\_cacheable\_read\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_normal\_wt\_cacheable\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal write-through write external aborts.

**ext\_abort\_normal\_wt\_cacheable\_write\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of normal write-through write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_normal\_cacheable\_write\_is\_sync.

**ext\_abort\_normal\_wt\_cacheable\_write\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_wt\_cacheable\_write\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_prefetch\_device\_GRE\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by device-GRE instruction fetches.

**ext\_abort\_prefetch\_device\_GRE\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by device-GRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_prefetch\_device\_nGRE\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by device-nGRE instruction fetches.

**ext\_abort\_prefetch\_device\_nGRE\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by device-nGRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_prefetch\_device\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by device-nGnRE instruction fetches.

**ext\_abort\_prefetch\_device\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by device-nGnRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_prefetch\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by normal writeback cacheable instruction fetches.

**ext\_abort\_prefetch\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Behaviour of external aborts generated by normal writeback cacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort.

**ext\_abort\_prefetch\_noncacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by normal noncacheable instruction fetches.

**ext\_abort\_prefetch\_noncacheable\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by normal noncacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_prefetch\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_prefetch\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_prefetch\_so\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by device-nGnRnE instruction fetches.

**ext\_abort\_prefetch\_so\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by device=nGnRnE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_prefetch\_wt\_cacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by normal writethrough cacheable instruction fetches.

**ext\_abort\_prefetch\_wt\_cacheable\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by normal writethrough cacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_so\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_so\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_so\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_so\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_so\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGnRnE write external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**ext\_abort\_so\_write\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].



**ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_ttw\_cacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of TTW cacheable read external aborts.

**ext\_abort\_ttw\_cacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of TTW cacheable read external aborts.

**ext\_abort\_ttw\_cacheable\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_ttw\_cacheable\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_ttw\_noncacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of TTW noncacheable read external aborts.

**ext\_abort\_ttw\_noncacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of TTW noncacheable read external aborts.

**ext\_abort\_ttw\_noncacheable\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_ttw\_noncacheable\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_ttw\_wt\_cacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of TTW write-through cacheable read external aborts.

**ext\_abort\_ttw\_wt\_cacheable\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of TTW write-through cacheable read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_ttw\_cacheable\_read\_is\_sync.

**ext\_abort\_ttw\_wt\_cacheable\_read\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_ttw\_cacheable\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_ttw\_wt\_cacheable\_read\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_ttw\_cacheable\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**external\_oslar\_access\_disabled\_by\_authentication****Type**

bool

**Default value**

0x0

**Description**

If true, external accesses to OSLAR, when external debugging is not enabled, will generate an error (FEAT\_Debugv8p2).

**fault\_on\_nT\_bit\_set****Type**

bool

**Default value**

0x1

**Description**

Whether block translation table entries with the nT bit set should always fault. Only applies when changing\_block\_size\_without\_bbm\_support\_level is 1 or higher.

**fault\_unalign\_to\_unsupported\_access****Type**

int

**Default value**

0x8

**Description**

If has\_unaligned\_single\_copy\_atomicity is true, whether unaligned A64 atomic, exclusive and acquire/release instructions to non iWB-oWB or the access crossing a 16-byte boundary generate fault. Bits 0,1,2,3 should be set accordingly to enable the fault behaviour. bit 0: atomic access should fault, bit 1: exclusive access should fault, bit 2: acquire/release should fault, bit 3: the access crossing a 16-byte boundary should fault.

**fault\_unaligned\_s1\_device\_s2\_fwb****Type**

int

**Default value**

0x0

**Description**

Whether unaligned fault with stage1 Device memory and final memory attribute forced to normal by FWB. 0, No fault. 1, will fault. 2 No fault if final Shareability is NSH.

**flash\_protection\_enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable flash memory protection after reset.

**`force_align_pc`****Type**

bool

**Default value**

0x0

**Description**

UNPREDICTABLE branch to non-word-aligned address in ARM state is forced to be aligned.

**`force_sync_on_wfx`****Type**

bool

**Default value**

0x0

**Description**

If true, the PE does a context synchronization before entering low power state(WFI/WFE).

**`fpcr_short_vector_raz`****Type**

bool

**Default value**

0x0

**Description**

FPSCR and FPCR fields LEN and STRIDE are hardwired to 0.

**`fpsr_res0_stateful_mask`****Type**

int

**Default value**

0x0

**Description**

Mask for stateful bits of FPSR which are RES0.

**`gic.GICC-offset`****Type**

int

**Default value**

0x2000

**Description**

Offset from PERIPHBASE for GICC registers.

**`gic.GICD-offset`****Type**

int

**Default value**

0x1000

**Description**

Offset from PERIPHBASE for GICD registers. Will be ignored when GICv3 CPU interface is enabled, as distributor is then external to the cluster.

**`gic.GICH-offset`****Type**

int

**Default value**

0x4000

**Description**

Offset from PERIPHBASE for GICH registers.

**`gic.GICH-other-CPU-offset`****Type**

int

**Default value**

0x5000

**Description**

Offset from PERIPHBASE for GICH registers for accessing other CPUs in the cluster. Set to 0 to disable.

**`gic.GICV-alias`****Type**

int

**Default value**

0x0

**Description**

Offset from PERIPHBASE for alias of GICV registers. When gicv2-only, if zero no alias will be created; if gicv2-only=0, the param is deprecated, when zero or unset an alias is created in the place mandated by the architecture (GICV-base+0xF000).

**gic.GICV-offset****Type**

int

**Default value**

0x6000

**Description**

Offset from PERIPHBASE for GICV registers.

**gic.PERIPH-size****Type**

int

**Default value**

0x8000

**Description**

Size of registers based at PERIPHBASE that are considered to be owned by the GIC. Any accesses in the range PERIPHBASE to PERIPHBASE+gic.PERIPH-size-1 that do not match GIC registers will be treated as RAZ/WI.

**gic\_iri.ARE-fixed-to-one****Type**

bool

**Default value**

0x1

**Description**

GICv2 compatibility is not supported and GICD\_CTLR.ARE\_\* is always one.

**gic\_iri.DPG-ARE-only****Type**

bool

**Default value**

0x0

**Description**

Limit application of DPG bits to interrupt groups for which ARE=1.

**gic\_iri.DPG-bits-implemented****Type**

bool

**Default value**

0x0

**Description**

Enable implementation of interrupt group participation bits or DPG bits in GICR\_CTLR.

**`gic_iri.GICD-alias`****Type**

int

**Default value**

0x0

**Description**

In GICv2 mode: the base address for a 4k page alias of the first 4k of the Distributor page, in GICv3 mode. the base address of a 64KB page containing message based SPI signalling register aliases(0:Disabled).

**`gic_iri.GICD_ITARGETSR-RAZWI`****Type**

bool

**Default value**

0x0

**Description**

If true, the GICD\_ITARGETS registers are RAZ/WI.

**`gic_iri.GICD_PIDR`****Type**

int

**Default value**

0x0

**Description**

The value for the GICD\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

**`gic_iri.GICR_PIDR`****Type**

int

**Default value**

0x0

**Description**

The value for the GICR\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.



**`gic_iri.GITS_PIDR`****Type**

int

**Default value**

0x0

**Description**

The value for the GITS\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

**`gic_iri.ICFGR-rsvd-bit`****Type**

bool

**Default value**

0x1

**Description**

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGRn for n>0.

**`gic_iri.IIDR`****Type**

int

**Default value**

0x0

**Description**

GICD\_IIDR and GICR\_IIDR value.

**`gic_iri.IRI-ID-bits`****Type**

int

**Default value**

0x10

**Description**

Number of bits used to represent interrupts IDs in the Distributor and Redistributors, forced to 10 when none of LPIs, extended SPIs or extended PPIs is supported.

**`gic_iri.ITS-count`****Type**

int

**Default value**

0x0

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**`gic_iri.SPI-count`****Type**

int

**Default value**

0x20

**Description**

Number of SPIs that are implemented.

**`gic_iri.SPI-message-based-support`****Type**

bool

**Default value**

0x1

**Description**

Distributor supports message based signaling of SPI.

**`gic_iri.STATUSR-implemented`****Type**

bool

**Default value**

0x0

**Description**

Determines whether the GICR\_STATUSR register is implemented.

**`gic_iri.enable_protocol_checking`****Type**

bool

**Default value**

0x0

**Description**

Enable/disable protocol checking at cpu interface.

**`gic_iri.enabled`****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**`gic_iri.has-two-security-states`****Type**

bool

**Default value**

0x0

**Description**

If true, has two security states.

**`gic_iri.irouter-default-mask`****Type**

string

**Default value**

0.0.0.7

**Description**

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

**`gic_iri.irouter-default-reset`****Type**

string

**Default value**

0.0.0.0

**Description**

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or '\*'.

**`gic_iri.monolithic`****Type**

bool

**Default value**

0x1

**Description**

Indicate that the implementation is not distributed.

**`gic_iri.non-ARE-core-count`****Type**

int

**Default value**

0x4

**Description**

Maximum number of non-ARE cores; normally used to pass the cluster-level NUM\_CORES parameter to the top-level redistributor.

**`gic_iri.periph-size`****Type**

int

**Default value**

0x0

**Description**

Size in bytes allocated to internal GIC Distributor.

**`gic_iri.priority-bits`****Type**

int

**Default value**

0x5

**Description**

Number of implemented priority bits.

**`gic_iri.processor-numbers`****Type**

string

**Default value**

""

**Description**

Specify processor numbers (as appears in GICR\_TYPER) in the form 0.0.0.0=0,0.0.0.1=1 etc.)  
If not specified, will number processors starting at 0.

**`gic_iri.redistributor-offset`****Type**

int

**Default value**

0x0

**Description**

Offset from reg-offset where the Redistributors are accessible.

**`gic_iri.redistributor-size`****Type**

int

**Default value**

0x0

**Description**

Per Redistributor register space in bytes.

**`gic_iri.reg-offset`****Type**

int

**Default value**

0x0

**Description**

Offset from PERIPHBASE allocated to internal GIC Distributor.

**`gic_iri.supports-shareability`****Type**

bool

**Default value**

0x0

**Description**

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

**`gic_iri.virtual-lpi-support`****Type**

bool

**Default value**

0x0

**Description**

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

**`gic_iri.wakeup-on-reset`****Type**

bool

**Default value**

0x0

**Description**

Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

**gicv3.A3-affinity-supported****Type**

bool

**Default value**

0x0

**Description**

Whether a non-zero value for affinity at level 3 is supported.

**gicv3.BPR-min****Type**

int

**Default value**

0x2

**Description**

The minimum value for the GICC\_BPR register (non-secure version will be 1 + this value).

**gicv3.EOI-check-CPUID****Type**

bool

**Default value**

0x0

**Description**

Check CPU ID specified for accesses to EOI registers (rather than just ending highest priority active interrupt).

**gicv3.EOI-check-ID****Type**

bool

**Default value**

0x0

**Description**

Check Interrupt ID specified for accesses to EOI registers (rather than just ending highest priority active interrupt).

**gicv3.EOI-deactivate-any-interrupt****Type**

bool

**Default value**

0x0

**Description**

Allow an EOI to deactivate interrupts that aren't the highest priority active interrupt (EOI-ignore-out-of-order must be false otherwise this is ignored).

**`gicv3.EOI-ignore-out-of-order`****Type**

bool

**Default value**

0x1

**Description**

Ignore EOI writes that cannot end the highest priority active interrupt.

**`gicv3.FIQEn-RAO`****Type**

bool

**Default value**

0x0

**Description**

GICC\_CTLR.FIQEn is read as one, write insensitive.

**`gicv3.IIDR_base`****Type**

int

**Default value**

0x43b

**Description**

The base value for calculating the GICC\_IIDR register value.

**`gicv3.LR-count`****Type**

int

**Default value**

0x10

**Description**

The number of implemented list registers.

**`gicv3.PMHE-RAO-WI`****Type**

bool

**Default value**

0x0

**Description**

ICC\_CTLR\_EL\*.PHME is read as one, write insensitive.

**gicv3.PMHE-RAZ-WI****Type**

bool

**Default value**

0x0

**Description**

ICC\_CTLR\_EL\*.PHME is read as zero, write insensitive.

**gicv3.PMHE-release-set-packet****Type**

bool

**Default value**

0x0

**Description**

if PHME is enabled, whether a SET packet is released by CPU Intf in Upstream Ack window.

**gicv3.SRE-EL2-enable-RAO****Type**

bool

**Default value**

0x0

**Description**

When ICC\_SRE\_EL2.SRE is RAO/WI, makes ICC\_SRE\_EL2.Enable RAO/WI.

**gicv3.SRE-EL3-enable-RAO****Type**

bool

**Default value**

0x0

**Description**

When ICC\_SRE\_EL3.SRE is RAO/WI, makes ICC\_SRE\_EL3.Enable RAO/WI.

**gicv3.SRE-EL3-set-once****Type**

bool



**Default value**

0x0

**Description**

Restrict SRE EL3 to be set only once.

**`gicv3.SRE-enable-action-on-mmap`****Type**

int

**Default value**

0x0

**Description**

Allowed values are: 0-SRE one allows mmap access. 1-SRE one disables mmap access. 2-SRE one makes mmap access RAZ-WI.

**`gicv3.STATUSR-implemented`****Type**

bool

**Default value**

0x1

**Description**

If GICv3 CPU interface is being used, this determines whether the STATUS registers are implemented.

**`gicv3.VBPR-min`****Type**

int

**Default value**

0x2

**Description**

The minimum value for the GICV\_BPR register (non-secure version will be 1 + this value).

**`gicv3.VFIQEn-RAO`****Type**

bool

**Default value**

0x0

**Description**

ICH\_VMCR\_EL2.VFIQEn is read as one, write insensitive.

**gicv3.cpuintf-mmap-access-level****Type**

int

**Default value**

0x0

**Description**

Allowed values are: 0-mmap access is supported for GICC,GICH,GICV registers. 1-mmap access is supported only for GICV registers. 2-mmap access is not supported.

**gicv3.dir-trap-support****Type**

bool

**Default value**

0x1

**Description**

The cpu supports separate trapping of ICC\_DIR\_EL1 to EL2.

**gicv3.el3\_trap\_priority\_when\_secure\_debug\_disabled****Type**

bool

**Default value**

0x0

**Description**

Undef to access priorities group register when secure debug is disabled.

**gicv3.extended-interrupt-range-support****Type**

bool

**Default value**

0x0

**Description**

Device has support for extended SPI/PPI ID ranges.

**gicv3.gicv2-only****Type**

bool

**Default value**

0x0

**Description**

Limit the GIC implementation to GICv2 features only.

**`gicv3.idle-is-ff`****Type**

bool

**Default value**

0x1

**Description**

For GICC/GICV RPR, when idle, return FF when true, minimum supported priority otherwise.

**`gicv3.ignore-DIR-write-when-EOImode-not-set`****Type**

bool

**Default value**

0x1

**Description**

Ignore UNPREDICTABLE access to GICC\_DIR register.

**`gicv3.interrupt-bypass-support`****Type**

bool

**Default value**

0x1

**Description**

Interrupt bypass support, set to false for devices not supporting interrupt bypass.

**`gicv3.local-SEIs`****Type**

bool

**Default value**

0x0

**Description**

Generate SEI to signal internal issues.

**`gicv3.local-VSEIs`****Type**

bool

**Default value**

0x0

**Description**

Generate VSEI to signal internal issues.

**`gicv3.physical-ID-bits`****Type**

int

**Default value**

0x10

**Description**

Number of physical ID bits implemented.

**`gicv3.priority-bits`****Type**

int

**Default value**

0x5

**Description**

Number of priority bits implemented.

**`gicv3.send-PMHE-command-only-when-priority-changes`****Type**

bool

**Default value**

0x0

**Description**

Send PMHE upstream command to distributor only when write to ICC\_PMR\_EL1 changes the priority.

**`gicv3.sgi-range-selector-support`****Type**

bool

**Default value**

0x0

**Description**

Device has support for the Range Selector feature for SGI.

**`gicv3.suppress-virtual-enables-comms`****Type**

bool

**Default value**

0x1

**Description**

In GICv3 only mode, prevents the GIC CPUIF from communicating UpstreamWrite/VirtualEnables to the IRI.

**`gicv3.virtual-ID-bits`****Type**

int

**Default value**

0x10

**Description**

Number of virtual ID bits implemented.

**`gicv3.virtual-lpi-support`****Type**

bool

**Default value**

0x1

**Description**

When GICv3 is supported, indicates a cut down CPUIF interface with no support of VLPI (GICv3 only) when false.

**`gicv3.virtual-priority-bits`****Type**

int

**Default value**

0x5

**Description**

Number of virtual priority bits implemented.

**`gicv3.without-DS-support`****Type**

bool

**Default value**

0x0

**Description**

GICv3 CPU interfaces do not support disabling security in the distributor (GICD\_CTLR.DS=1).

**gicv4.mask-virtual-interrupt****Type**

bool

**Default value**

0x0

**Description**

If true, virtual interrupts can be masked from being reported to virtual CPU interface by setting ICH\_HCR\_EL2.DVIM 1. No control otherwise.

**gicv5.config\_file****Type**

string

**Default value**

""

**Description**

File path for the GICv5 configuration yaml. The file lists the GICv5 params.

**gicv5.interrupt-bypass-support****Type**

bool

**Default value**

0x0

**Description**

Interrupt bypass support. when set to true, bypasses GICv5 CPU interface to signal interrupts to the PE.

**global\_debug\_rom.ROMDEVID****Type**

int

**Default value**

0x0

**Description**

Value of Debug Rom Device Identification Register.

**global\_debug\_rom.ROMPIDR****Type**

int

**Default value**

0x4000bb000

**Description**

Value of Debug Rom Peripheral Identification Register.

**global\_debug\_rom.ROMPRIDR0****Type**

int

**Default value**

0x1

**Description**

Value of Debug ROM Power RequestID Register.

**hardware\_translation\_table\_update\_implemented****Type**

int

**Default value**

0x1

**Description**

Implement hardware translation table updates from ARMv8R-64. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has-gicv4.1****Type**

bool

**Default value**

0x0

**Description**

GICv4.1 is enabled, and all the features with GICv4.1 are implemented (FEAT\_GICv4p1).

**has\_16bit\_asids****Type**

bool

**Default value**

0x1

**Description**

Enable 16-bit ASIDs.

**has\_16bit\_vmids****Type**

int

**Default value**

0x1

**Description**

Implement support for 16-bit VMIDs from ARMv8R-64. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_16k\_granule****Type**

bool

**Default value**

0x0

**Description**

Implement the 16k LPAE translation granule.

**has\_4k\_granule****Type**

bool

**Default value**

0x1

**Description**

Implement the 4k LPAE translation granule.

**has\_64k\_granule****Type**

bool

**Default value**

0x1

**Description**

Implement the 64k LPAE translation granule.

**has\_aarch32\_dbgdidr\_etc****Type**

bool

**Default value**

0x1

**Description**

DBGDIDR, DBGDRAR, DBGDSAR exist even if EL1 doesn't implement AArch32.



**has\_aarch64****Type**

bool

**Default value**

0x0

**Description**

All implemented exception levels can run in AArch64.

**has\_bc****Type**

int

**Default value**

0x1

**Description**

Implement Armv8.8 Hinted Conditional Branch (FEAT\_HBC) Possible values of this parameter are: - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_ccidx****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8R-64 CCSIDR Extension. Extending the ccsidr number of sets (FEAT\_CCIDX).

**has\_cluster\_l1cache\_size****Type**

bool

**Default value**

0x1

**Description**

Whether core supports cluster level l1cache size.

**has\_coherent\_icache****Type**

bool

**Default value**

0x0

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_const\_pac****Type**

int

**Default value**

0x0

**Description**

Feature for singular selection of PAC field (FEAT\_CONSTPACFIELD). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_cvadp\_support****Type**

int

**Default value**

0x0

**Description**

Implement instruction to support cache clean by deep persistence (DC CVADP) from ARMv8.5, can be selected for core implemented on any arch version starting ARMv8.2 (FEAT\_DPB, FEAT\_DPB2). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_debug\_rom****Type**

bool

**Default value**

0x1

**Description**

If true, a debug ROM will be generated describing the cluster's debug components.

**has\_delayed\_ctireg****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of CTI register writes until ISB or implicit barrier.

**has\_delayed\_dbgreg****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

**has\_delayed\_mdscr\_el1****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of MDSCR\_EL1 register writes until ISB or implicit barrier.

**has\_delayed\_oslar\_el1****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of OSLAR\_EL1 register writes until ISB or implicit barrier.

**has\_delayed\_pmureg****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of PMU register writes until ISB or implicit barrier.

**has\_delayed\_sysreg****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of system register writes until ISB or implicit barrier.

**has\_dgh****Type**

int

**Default value**

0x1

**Description**

Implements Data Gathering Hint instruction from ARMv8.6 (FEAT\_DGH). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**has\_e0pd****Type**

int

**Default value**

0x0

**Description**

Implement ARMv8-R64 feature to prevent unprivileged access to one half of the memory  
Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_edacr****Type**

bool

**Default value**

0x1

**Description**

Implement EDACR register.

**has\_enhanced\_pac****Type**

bool

**Default value**

0x0

**Description**

If pointer authentication is enabled then implement enhanced PAC.

**has\_exception\_trapping\_form\_of\_vector\_catch****Type**

bool

**Default value**

0x1

**Description**

Implement the exception trapping form of vector catch debug event.

**has\_export\_m\_port****Type**

bool

**Default value**

0x1

**Description**

The interrupt distributor has an optional interrupt export port for routing interrupts to an external device.

**has\_far\_not\_valid****Type**

bool

**Default value**

0x0

**Description**

Implements FnV bit in ESR\_ELx and xFSR, FAR not valid for synchronous external aborts.

**has\_far\_not\_valid\_dfsc****Type**

bool

**Default value**

0x0

**Description**

Implements FnV bit in ESR\_ELx, FAR not valid for synchronous external aborts for Data Abort.

**has\_far\_not\_valid\_ifsc****Type**

bool

**Default value**

0x0

**Description**

Implements FnV bit in ESR\_ELx and xFSR, FAR not valid for synchronous external aborts for Instruction Abort.

**has\_flash****Type**

bool

**Default value**

0x0

**Description**

Flash Port present.

**has\_flash\_protection****Type**

bool

**Default value**

0x0

**Description**

Implement flash memory protection.

**has\_fp16****Type**

int

**Default value**

0x1

**Description**

Implement the half-precision floating-point data processing instructions from ARMv8R-64 (FEAT\_FP16). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_generic\_authentication****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.3 generic authentication. Possible values of this parameter are: - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_hardware\_translation\_table\_update****Type**

int

**Default value**

0x2

**Description**

Type of hardware translation table supported (when enabled by `hardware_translation_table_update_implemented`). 0, not implemented. 1, access bit updates implemented. 2, access bit updates and dirty bit mechanism implemented.

**`has_internal_gic_iri`****Type**

bool

**Default value**

0x0

**Description**

Is Internal GIC IRI implemented.

**`has_itd`****Type**

bool

**Default value**

0x1

**Description**

Implement the optional IT disable feature.

**`has_large_system_ext`****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8 Large System Extensions (FEAT\_LSE).

**`has_large_va`****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8R-64 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_llpp****Type**

bool

**Default value**

0x0

**Description**

Low Latency Peripheral Port present.

**has\_mpm****Type**

bool

**Default value**

0x0

**Description**

Implement max-power mitigation mechanism (MPMM).

**has\_no\_os\_double\_lock****Type**

int

**Default value**

0x0

**Description**

Do not implement the OS double-lock (FEAT\_DoubleLock). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_par\_bit10\_razwi****Type**

bool

**Default value**

0x0

**Description**

Whether PAR\_EL1[10] is RAZ/WI.

**has\_partial\_delayed\_mdscr\_el1****Type**

bool

**Default value**

0x0



**Description**

has\_delayed\_oslar\_el1 only apply to some bits of MDSCR\_EL1 (MDE, KDE, TDCC, SS).

**has\_pc\_sample\_based\_profiling****Type**

bool

**Default value**

0x1

**Description**

If true, pc sample-based profiling is enabled (FEAT\_PCSRv8, FEAT\_PCSRv8p2).

**has\_per\_cluster\_debug\_auth\_ports****Type**

bool

**Default value**

0x0

**Description**

If true then the debug authentication ports i.e. spniden, niden, rpliden, rtpiden, dbgen, spiden are available per cluster.

**has\_p12****Type**

bool

**Default value**

0x1

**Description**

Whether EL2 is implemented.

**has\_pmc****Type**

bool

**Default value**

0x0

**Description**

Programmable MBIST controllers implemented.

**has\_pmu****Type**

int

**Default value**

0x1

**Description**

Implement the optional Performance Monitors Extension (FEAT\_PMUv3). 0, Not Implemented. 1, Implemented. 2, PMU is IMPLEMENTATION\_DEFINED, PMU version would be set to 0xF and would behave as if no PMU is implemented.

**has\_pointer\_authentication****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.3 pointer authentication (FEAT\_PAuth). Possible values of this parameter are: - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_prediction\_invalidation\_instructions****Type**

int

**Default value**

0x1

**Description**

Implement execution and data prediction invalidation from ARMv8-R64 (FEAT\_SPECRES). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_qarma3\_pac****Type**

bool

**Default value**

0x0

**Description**

Supports QARMA3 pointer authentication algorithm (FEAT\_PACQARMA3).

**has\_ras****Type**

int

**Default value**

0x0

**Description**

Implements the ARMv8 RAS Extension. 0 = NO\_RAS, 1 = MINIMAL\_RAS, 2 = FULL\_RAS (FEAT\_RAS).

**has\_ras\_armv84\_extension****Type**

int

**Default value**

0x0

**Description**

Implement ARMv8R-64 RAS Extension (FEAT\_RASv1p1). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_ras\_double\_fault****Type**

int

**Default value**

0x0

**Description**

Implement ARMv8.4 RAS Double Fault Extension (FEAT\_DoubleFault). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_restriction\_on\_speculative\_data\_loaded****Type**

int

**Default value**

0x1

**Description**

Implements the ARMv8-R64 security feature (Restrictions on the effects of speculation) (FEAT\_CSV3). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_self\_hosted\_trace\_extension****Type**

int

**Default value**

0x1

**Description**

Implement support for the Self-hosted Trace Extensions from ARMv8R-64 (FEAT\_TRF).

Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_small\_page\_table****Type**

int

**Default value**

0x1

**Description**

Implement small page table support which increases the maximum value of TxSZ field from ARMv8R-64 (FEAT\_TTST). Note: will be unimplemented only if both has\_small\_page\_table=0x0 and has\_pl2=0x0. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_software\_lock****Type**

bool

**Default value**

0x1

**Description**

Implement software lock in memory-mapped CTI, PMU, and external debug interfaces.

**has\_speculation\_barrier\_inst****Type**

int

**Default value**

0x1

**Description**

Implement speculation barrier instruction (SB) from ARMv8-R64 (FEAT\_SB). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_speculative\_sei****Type**

bool

**Default value**

0x0

**Description**

If true, the PE can generate SError interrupt exceptions from speculative reads of memory, including speculative instruction fetches.

**has\_spp****Type**

bool

**Default value**

0x0

**Description**

Shared Peripheral Port present.

**has\_stage2\_ap\_speculative\_update****Type**

int

**Default value**

0x0

**Description**

Speculative update of S2 AP bit on S1 TTW. 0 = No Update, 1 = Update, 2 = Update including for AT ops.

**has\_synchronous\_load\_atomics****Type**

bool

**Default value**

0x1

**Description**

Report asynchronous abort due to unsupported load atomics as synchronous (Cacheable).

**has\_synchronous\_load\_atomics\_noncacheable****Type**

bool

**Default value**

0x1

**Description**

Report asynchronous abort due to unsupported load atomics as synchronous (Non-Cacheable).

**has\_synchronous\_store\_atomics****Type**

bool

**Default value**

0x0

**Description**

Report asynchronous abort due to unsupported store atomics as synchronous (Cacheable).

**has\_synchronous\_store\_atomics\_noncacheable****Type**

bool

**Default value**

0x0

**Description**

Report asynchronous abort due to unsupported store atomics as synchronous (Non-Cacheable).

**has\_tlb\_conflict\_abort****Type**

bool

**Default value**

0x0

**Description**

Detected inconsistent TLB content generate aborts.

**has\_tlb\_pa\_caching****Type**

bool

**Default value**

0x0

**Description**

Whether intermediate caching of translation table walks might include NonCoherent caches of previous valid walks. 0, NonCoherent caches might be included. 1, No NonCoherent caches included (FEAT\_nTLBPA).

**has\_unsupported\_exclusive\_fault****Type**

bool

**Default value**

0x1

**Description**

Report unsupported exclusive access with Unsupported Exclusive fault status (otherwise use external abort).

**has\_v8\_4\_pmu\_extension****Type**

int

**Default value**

0x1

**Description**

Implement PMU extension from ARMv8.4 (FEAT\_PMUv3p4). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_v8\_5\_debug\_over\_power\_down****Type**

int

**Default value**

0x0

**Description**

Implement ARMv8.5 Debug over powerdown Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8R-64 is enabled. - 2, feature is implemented.

**has\_v8\_6\_pmu\_events****Type**

int

**Default value**

0x1

**Description**

Implements PMU events from ARMv8.6 Possible values of this parameter are: - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**has\_writebuffer****Type**

bool

**Default value**

0x0

**Description**

Implement write accesses buffering before L1 cache. May affect ext\_abort behaviour.

**hcptr\_tta\_behaviour****Type**

int

**Default value**

0x2

**Description**

Behaviour of HCPTR.TTA when there is no CP14 ETM interface. 0, RAZ/WI. 1, RAO/WI. 2, stateful.

**hcr\_el2\_miocnce\_is\_rw****Type**

bool

**Default value**

0x0

**Description**

If true, HCR\_EL2.MIOCNCNCE is treated as R/W instead of RAZ/WI.

**hcr\_swio\_res1****Type**

bool

**Default value**

0x0

**Description**

Whether HCR.SWIO and/or HCR\_EL2.SWIO are RES1.

**hsr\_uncond\_cc****Type**

bool

**Default value**

0x0

**Description**

Condition codes reported in HSR as AL if it passes.

**icache-hit\_latency****Type**

int

**Default value**

0x0



**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**`icache-log2linelen`****Type**

int

**Default value**

0x0

**Description**

If nonzero, Log2 of the instruction cache line length in bytes (valid values in range 4-8). Otherwise the value of `cache-log2linelen` is used.

**`icache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-nprefetch`****Type**

int

**Default value**

0x1

**Description**

Number of next sequential instruction cache lines to prefetch. This is only used when `icache-prefetch_enabled=true`.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-prefetch\_level****Type**

int

**Default value**

0x0

**Description**

0 based cache level at which instructions are pre-fetched. This is only used when icache-prefetch\_enabled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x8

**Description**

L1 I-Cache read bus width in bytes used to calculate per-access timing annotations.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**icache-ways****Type**

int

**Default value**

0x2

**Description**

L1 I-Cache number of ways (sets are implicit from size).

**ignore\_tag\_check\_dcc\_load\_store\_in\_ma\_mode\_when\_tco\_is\_disabled****Type**

bool

**Default value**

0x0

**Description**

Constrained unpredictable behavior for reads/writes to external debug interface DTR regs in memory access mode when PSTATE.TCO is 0. If true, tag check is ignored else, tag check is performed if required.

**`imp_def_functionality_behaviour`****Type**

int

**Default value**

0x0

**Description**

Behaviour of IMPLEMENTATION DEFINED registers and system instructions. 0, UNDEF. 1, RAZ/WI.

**`instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**`internal_vgic`****Type**

bool

**Default value**

0x0

**Description**

Instantiate VGIC peripheral in this processor.

**`is_debug_state_pmu_snapshot_allowed`****Type**

bool

**Default value**

0x1

**Description**

If true, PMU snapshot is allowed in debug state.

**is\_first\_pcsr\_sample\_ignored****Type**

bool

**Default value**

0x0

**Description**

If true, First read of PMPCSR register after reset returns 0xFFFFFFFF.

**is\_serror\_edge\_triggered****Type**

bool

**Default value**

0x1

**Description**

If true, SError is edge-triggered. Otherwise, its level-triggered.

**is\_uniprocessor****Type**

bool

**Default value**

0x0

**Description**

Value for the U bit in MPIDR. true disables L1 cache coherency protocols.

**isb\_is\_branch****Type**

bool

**Default value**

0x0

**Description**

If true, ISB is considered an immediate branch. This allows to count ISB as a branch in BRBE.

**ish\_is\_osh****Type**

bool

**Default value**

0x0

**Description**

Whether Innershareable is same as OuterShareable.

**itd\_conditional\_instructions\_are\_32bit****Type**

bool

**Default value**

0x0

**Description**

When SCTLR\_ELx.ITD=1, an IT instruction plus a T16 instruction are considered a single 32bit conditional instruction.

**jidr\_is\_undef\_at\_el0****Type**

bool

**Default value**

0x0

**Description**

If true, JIDR register access is UNDEF at EL0.

**jmcr\_is\_undef\_at\_el0****Type**

bool

**Default value**

0x0

**Description**

If true, JMCR register access is UNDEF at EL0.

**joscr\_is\_undef\_at\_el0****Type**

bool

**Default value**

0x0

**Description**

If true, JOSCR register access is UNDEF at EL0.

**l1cache\_has\_r52\_cache\_policy****Type**

bool

**Default value**

0x0

**Description**

Whether l1cache has r52 cache policy.

**l1cache\_has\_rsvd\_flash\_ways****Type**

bool

**Default value**

0x0

**Description**

Whether l1 cache segregates ways for flash and AXI data.

**l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l2cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x8

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.



**l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l2cache-ways****Type**

int

**Default value**

0x10

**Description**

L2 Cache number of ways (sets are implicit from size).

**l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x8

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x0

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-mpamf.arch\_major\_ver****Type**

int

**Default value**

0x0

**Description**

L3 Cache MPAMF\_AIDR architecture major version.

**l3cache-mpamf.arch\_minor\_ver****Type**

int

**Default value**

0x0

**Description**

L3 Cache MPAMF\_AIDR architecture minor version.

**l3cache-mpamf.esr\_mask****Type**

int

**Default value**

0xffffffff

**Description**

L3 Cache MPAMF\_ESR mask value.

**l3cache-mpamf.has\_esr****Type**

bool

**Default value**

0x0

**Description**

L3 Cache's MPAMF\_ESR, MPAMF\_ECR, and MPAM error handling implemented.

**13cache-mpamf.has\_extd\_esr****Type**

bool

**Default value**

0x0

**Description**

L3 Cache's MPAMF\_ESR is 64-bits.

**13cache-mpamf.has\_impl\_idr****Type**

bool

**Default value**

0x0

**Description**

L3 Cache's MPAMF\_IMPL\_IDR is present.

**13cache-mpamf.has\_mbwu\_long\_counter****Type**

bool

**Default value**

0x0

**Description**

L3 Cache has long MBWU counter and capture registers.

**13cache-mpamf.has\_mpamfidr\_ext****Type**

bool

**Default value**

0x0

**Description**

MPAMF\_IDR.EXT support.

**13cache-mpamf.has\_partid\_nrw****Type**

bool

**Default value**

0x0

**Description**

Narrowing part ID register is present. This is global rather than per-instance.

**l3cache-mpamf.has\_priority\_partitioning****Type**

bool

**Default value**

0x0

**Description**

The selected resource has priority partitioning described in MPAMF\_PRI\_IDR.

**l3cache-mpamf.has\_prod\_id****Type**

int

**Default value**

0x0

**Description**

L3 Cache MPAMF\_IIDR product ID supported.

**l3cache-mpamf.has\_prod\_rev****Type**

int

**Default value**

0x0

**Description**

L3 Cache MPAMF\_IIDR product REVISION supported.

**l3cache-mpamf.has\_prod\_var****Type**

int

**Default value**

0x0

**Description**

L3 Cache MPAMF\_IIDR product VARIANT supported.

**l3cache-mpamf.has\_ris****Type**

bool

**Default value**

0x0

**Description**

L3 Cache has resource instance selection support.

**l3cache-mpamf.max\_partid\_ns****Type**

int

**Default value**

0xffff

**Description**

L3 Cache Maximum value of non-secure PARTID supported.

**l3cache-mpamf.max\_partid\_rl****Type**

int

**Default value**

0xffff

**Description**

L3 Cache Maximum value of realm PARTID supported for RME implementations.

**l3cache-mpamf.max\_partid\_rt****Type**

int

**Default value**

0xffff

**Description**

L3 Cache Maximum value of root PARTID supported for RME implementations.

**l3cache-mpamf.max\_partid\_s****Type**

int

**Default value**

0xffff

**Description**

L3 Cache Maximum value of secure PARTID supported.

**l3cache-mpamf.max\_pmg\_ns****Type**

int

**Default value**

0xff

**Description**

L3 Cache Maximum value of non-secure PMG supported.

**l3cache-mpamf.max\_pmg\_r1****Type**

int

**Default value**

0xff

**Description**

L3 Cache Maximum value of realm PMG supported for RME implementations.

**l3cache-mpamf.max\_pmg\_rt****Type**

int

**Default value**

0xff

**Description**

L3 Cache Maximum value of root PMG supported for RME implementations.

**l3cache-mpamf.max\_pmg\_s****Type**

int

**Default value**

0xff

**Description**

L3 Cache Maximum value of secure PMG supported.

**l3cache-mpamf.mbwu\_long\_counter\_width****Type**

int

**Default value**

0x0

**Description**

L3 Cache long MBWU counter width. 0: 63 bits, 1: 44 bits.

**l3cache-mpamf.no\_impl\_msmon****Type**

bool

**Default value**

0x0

**Description**

L3 Cache's MPAMF\_IMPL\_IDR does not describe resource monitors.

**l3cache-mpamf.no\_impl\_part****Type**

bool

**Default value**

0x0

**Description**

L3 Cache's MPAMF\_IMPL\_IDR does not describe resource partitioning controls.

**l3cache-mpamf.ris\_max****Type**

int

**Default value**

0x0

**Description**

L3 Cache's largest resource instance selector value defined.

**l3cache-mpamf\_base****Type**

int

**Default value**

0x0

**Description**

L3 Cache memory mapped MPAM registers base address.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.



**l3cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x8

**Description**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x0

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**l3cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x8

**Description**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**legacy\_combining\_exc\_catch\_trace****Type**

bool

**Default value**

0x1

**Description**

Whether exception catch is traced as part of exception entry/exit in same cycle.

**ls64\_ignore\_s1\_unpred\_memattr\_transformation****Type**

bool

**Default value**

0x0

**Description**

If true, stage 1 unpredictable memory attribute transformations are ignored for FEAT\_LS64 single-copy atomic 64-byte load/store instructions' (FEAT\_LS64, FEAT\_LS64\_V, FEAT\_LS64\_ACCDATA).

**ls64\_memtype\_check\_use\_combined\_memattr****Type**

int

**Default value**

0x0

**Description**

FEAT\_LS64 single-copy atomic 64-byte load/store instructions' 0 : memory attributes check is performed at each enabled stage of translation, 1 : memory attributes check is done on the combined memory attributes only. 2. memory attributes check is done on the combined memory attributes with Stage1 and Stage2 fault get evaluated to check on which stage fault should be reported.

**ls64wb\_memtype\_check\_allow\_any\_cacheable\_memattr****Type**

bool

**Default value**

0x0

**Description**

If true, when FEAT\_LS64WB is implemented, any cacheable memory access performed by LD/ST64B instructions is 64-byte, single-copy atomic.

**mdrar\_el1\_res0****Type**

bool

**Default value**

0x0

**Description**

MDRAR\_EL1 is RES0.

**memory.ext\_slave\_base****Type**

int

**Default value**

0x0

**Description**

Base address of Slave Port. Each core's region will offset by ext\_slave\_size\_per\_core.

**memory.ext\_slave\_size\_per\_core****Type**

int

**Default value**

0x0

**Description**

Size of Slave region for each core.

**memory.flash\_base****Type**

int

**Default value**

0x0

**Description**

Base address of Flash.

**memory.flash\_size****Type**

int

**Default value**

0x0

**Description**

Size of the Flash RAM.

**`memory.has_llram`****Type**

bool

**Default value**

0x0

**Description**

Low-Latency RAM present.

**`memory.l2_cache.is_inner_cacheable`****Type**

bool

**Default value**

0x1

**Description**

L2 cache obeys inner cacheable attributes (rather than outer cacheable attributes).

**`memory.l2_cache.is_inner_shareable`****Type**

bool

**Default value**

0x1

**Description**

L2 cache obeys inner shareable attributes (rather than outer shareable attributes).

**`memory.llram_base`****Type**

int

**Default value**

0x0

**Description**

Base address of LLRAM.

**`memory.llram_enable_at_reset`****Type**

bool

**Default value**

0x0

**Description**

Whether llram is enabled at reset.

**memory.llram\_shared****Type**

bool

**Default value**

0x0

**Description**

Controls the Low-Latency RAM's sharability attribute.

**memory.llram\_size****Type**

int

**Default value**

0x0

**Description**

Size of the LLRAM.

**memory.scu\_present****Type**

bool

**Default value**

0x1

**Description**

L1 Caches are coherent.

**memory.transmit\_vmid\_in\_user\_flags****Type**

bool

**Default value**

0x0

**Description**

Transmit VMID in transaction attributes.

**mixed\_endian****Type**

int

**Default value**

0x1

**Description**

Implement support for mixed endianness. 0, not supported. 1, supported at all exception levels. 2, supported at ELO only.

**`mpidr_layout`****Type**

int

**Default value**

0x0

**Description**

Layout of MPIDR. 0 AFF0 is CPUID, 1 AFF1 is CPUID.

**`non_secure_vgic_alias_when_ns_only`****Type**

int

**Default value**

0x0

**Description**

If ! has\_el3 and only non-secure side exists, then the normal position of the VGIC is a secure alias. If this parameter is non-zero then in addition a non-secure alias of the VGIC will be placed at this position (aligned to 32 KB).

**`num_protection_regions_s1`****Type**

int

**Default value**

0x20

**Description**

Number of v8-R protection regions.

**`num_protection_regions_s2`****Type**

int

**Default value**

0x20

**Description**

Number of v8-R hyp protection regions.

**num\_spi****Type**

int

**Default value**

0x20

**Description**

Number of interrupts (SPI) into the internal GIC controller.

**number\_of\_error\_records****Type**

int

**Default value**

0x0

**Description**

Cores Number of Error records supported for RAS.

**page\_based\_hardware\_attributes****Type**

int

**Default value**

0x0

**Description**

Implement the page based hardware attributes from ARMv8R-64. This parameter indicates which page table bits are available for hardware, where bits[3:0] correspond to PTE[62:59] and to TCR\_ELx.HWUnyy (FEAT\_HPDS2).

**par\_ns\_set\_unknown\_bit****Type**

bool

**Default value**

0x1

**Description**

Whether NS bit of PAR is set/clear when executing AT to perform non-secure regime translation. When true, NS is set to 1 else 0.

**par\_nse\_set\_unknown\_bit****Type**

bool

**Default value**

0x0



**Description**

Whether NSE bit of PAR is set/clear when executing AT operation on secure, non-secure or realm translation regime. When true, NSE is set to 1 else 0.

**per\_core\_master\_supported****Type**

bool

**Default value**

0x0

**Description**

If master port from each core is exposed out of cluster.

**pfr1\_csv2\_frac****Type**

int

**Default value**

0x0

**Description**

Fractional revision number ID\_AA64PFR1\_EL1.CSV2\_frac when ID\_AA64PFR0\_EL1.CSV==1 for CSV2 extension (FEAT\_CSV2\_1p1, FEAT\_CSV2\_1p2).

**pmb\_idr\_external\_abort****Type**

int

**Default value**

0x0

**Description**

Describes how the PE manages External aborts on writes made by the Statistical Profiling Extension to the Profiling Buffer. 0, External abort is reported to SPE, From Armv8.8 and Armv9.3, the value 0 is not permitted. 1, External abort is ignored. 2, The External abort generates an SError and the error is not reported to SPE.

**pmb\_idr\_flag\_updates****Type**

bool

**Default value**

0x1

**Description**

Defines whether the address translation performed by the Profiling Buffer manages the Access Flag and dirty state.

**pmbsr\_reports\_external\_abort****Type**

bool

**Default value**

0x1

**Description**

Whether PMBSR\_ELx.EA and PMBSR\_ELx.DL are set as the result of an external abort or are treated as RES0.

**pmcr\_disable\_events\_export****Type**

bool

**Default value**

0x1

**Description**

If true, export for PMU events is disabled. This configures PMCFGR.EX field.

**pmmir\_el1\_bus\_slots****Type**

int

**Default value**

0x0

**Description**

Largest value by which BUS\_ACCESS can increment over BUS\_CYCLES cycles.

**pmmir\_el1\_bus\_width****Type**

int

**Default value**

0x0

**Description**

Width, in bytes, of accesses counted by BUS\_ACCESS.

**pms\_idr\_max\_size****Type**

int

**Default value**

0x6

**Description**

Defines largest size for a single SPE record (rounded up to a power of 2).

**pmu-num\_counters****Type**

int

**Default value**

0x8

**Description**

Number of PMU counters implemented.

**pmu\_cycle\_counter\_counts\_actual\_cycles****Type**

bool

**Default value**

0x0

**Description**

If true and Timing annotation is enabled, PMU cycle counter counts actual cycles, otherwise counts instructions executed.

**pmu\_has\_chain\_event****Type**

bool

**Default value**

0x1

**Description**

PMU (if present) implements event number 0x1e, CHAIN.

**pseudo\_fault\_generation\_feature\_register****Type**

string

**Default value**

""

**Description**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_pseudo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED)

and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

**pstate\_ssbs\_type****Type**

int

**Default value**

0x0

**Description**

Implement speculative store bypass safe feature from ARMv8.5. 0, Not supported. 1, Supported without MSR/MRS access to SSBS (FEAT\_SSBS). 2, fully supported (FEAT\_SSBS2).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**ram\_protection\_enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable TCM, L1Cache memory protection after reset.

**randomize\_unknowns\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Will fill in unknown bits in registers at reset with random value using register\_reset\_data as seed, it overrides scramble\_unknowns\_at\_reset.

**ras\_extra\_configurations****Type**

string

**Default value**

""

**Description**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCN masks - these are 64 bit masks covering the 64 bit registers ERXMISCN\_EL1. E.g. [{"Index": 0, "EXMISCO\_mask": 0x0, "EXMISCO\_reset": 0x0, "EXMISC1\_mask": 0x0, "EXMISC1\_reset": 0x0, "EXMISC2\_mask": 0x0, "EXMISC2\_reset": 0x0, "EXMISC3\_mask": 0x0, "EXMISC3\_reset": 0x0, "EXCTLR\_EL1\_mask": 0x0, "EXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "EXMISCO\_mask": 0x0, "EXMISCO\_reset": 0x0, "EXSTATUS\_IERR\_mask": 0x300}].

**ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0x18

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**register\_reset\_data****Type**

int

**Default value**

0x0

**Description**

Data used to fill register bits when they become UNKNOWN at reset.

**register\_reset\_data\_hi****Type**

int

**Default value**

0x0

**Description**

Data used to fill the upper-half of 128-bit registers when the bits become UNKNOWN at reset.

**report\_iside\_cmo\_ifsr****Type**

bool

**Default value**

0x1

**Description**

fault info for an iside cache maintenance operation is reported in the IFSR.

**report\_second\_access\_align\_fault\_non\_atomic\_pair\_access****Type**

bool

**Default value**

0x0

**Description**

If true, the non-atomic load/store pair accesses report the 2nd register as faulting address for an alignment fault. This is IMP-DEF behavior as defined in FEAT\_LRCPC3.

**report\_second\_access\_mmu\_fault\_non\_atomic\_pair\_access****Type**

bool

**Default value**

0x0

**Description**

If true, the non-atomic load/store pair accesses report the 2nd register as faulting address for an MMU fault. This is IMP-DEF behavior as defined in FEAT\_LRCPC3.

**reported\_fp\_revision****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

**reported\_patch\_level****Type**

int

**Default value**`0xffffffffffffffff`**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**`int`**Default value**`0xffffffffffffffff`**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reserved\_HMC\_SSC\_PAC\_treated\_disabled****Type**`bool`**Default value**`0x0`**Description**

When `DBG[B|W]CR.{HMC,SSC,PAC}` bits configuration is reserved, this parameter controls whether breakpoints/watchpoints are treated as Disabled or not.

**restore\_fpsr\_on\_trapped\_fp\_exception****Type**`bool`**Default value**`0x0`**Description**

If true, FPSR is restored to the value of the FPSR immediately before the instruction that generated the trapped floating-point exception.

**restriction\_on\_speculative\_execution****Type**`int`**Default value**`0x0`

**Description**

Implements the ARMv8-R64 security feature (Restrictions on the effects of speculation): 0: No disclosure whether branch targets trained in one context can affect speculative execution in a different context, 1: Branch targets trained in one context cannot affect speculative execution in a different hardware described context (SCXTNUM\_ELx not supported), 2: Branch targets trained in one context cannot affect speculative execution in a different hardware described context (SCXTNUM\_ELx supported) (FEAT\_CSV2, FEAT\_CSV2\_2).

**rmr\_always\_implemented****Type**

bool

**Default value**

0x0

**Description**

Always implement RMR\_ELx, RMR, or HRMR at the highest implemented exception level, even if that exception level cannot use both AArch32 and AArch64.

**s1\_align\_memtype\_fault\_prio\_more\_than\_s2\_perm\_fault\_on\_s1\_walk****Type**

bool

**Default value**

0x1

**Description**

If true, s1 alignment fault has priority over s2 permission faults.

**s1\_perm\_fault\_prio\_more\_than\_s2\_perm\_fault\_on\_s1\_walk****Type**

bool

**Default value**

0x0

**Description**

If true, s1 permission fault has priority over s2 on s1 translation table walk permission faults.

**scheduler\_mode****Type**

int

**Default value**

0x0

**Description**

Control the interleaving of instructions in this processor. 0, default long quantum. 1, low latency mode, short quantum and signal checking. 2, lock-breaking mode, long quantum with



additional context switches near load-exclusive instructions. WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

**scr\_nET\_writeable****Type**

bool

**Default value**

0x0

**Description**

Whether SCR.nET is writeable. Writing to it is purely cosmetic (nET behavior not implemented).

**scramble\_unknowns\_at\_reset****Type**

bool

**Default value**

0x1

**Description**

Will fill in unknown bits in registers at reset with register\_reset\_data.

**seerror\_clear\_delay****Type**

int

**Default value**

0x0

**Description**

Delay for clearing of SError if SError is level-triggered, in cpu cycles.

**skip\_trace\_on\_write\_to\_oseccr\_el1\_when\_oslock\_is\_unlocked****Type**

bool

**Default value**

0x0

**Description**

If OSLSR\_EL1.OSLK == 0, then OSECCR\_EL1 returns an unknown value on reads and ignores writes. When true, also skips the traces on writes to OSECCR\_EL1 when OSLSR\_EL1.OSLK == 0.

**spp.base****Type**

int

**Default value**

0x0

**Description**

Sets the base address of Shared Peripheral Port.

**spp.size****Type**

int

**Default value**

0x1000

**Description**

Sets the size of SPP(in bytes).

**spsr\_el3\_is\_mapped\_to\_spsr\_mon****Type**

bool

**Default value**

0x0

**Description**

Whether SPSR\_EL3 is mapped to AArch32 register SPSR\_mon.

**spsr\_m4\_res0****Type**

bool

**Default value**

0x0

**Description**

Whether SPSR\_ELx.M[4] bit should be RES0 for AARCH64 only implementations.

**stage12\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

If VMSA is supported at stage1, number of stage1+2 tlb entries. If instruction\_tlb\_size !=0, this is treated as dtlb size.

**stage1\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1 only tlb entries.

**stage1\_walkcache\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1 only walk cache entries.

**strex\_fail\_can\_hit\_watchpoint****Type**

bool

**Default value**

0x0

**Description**

If true, a strex fail can hit watchpoint.

**supports\_multi\_threading****Type**

bool

**Default value**

0x0

**Description**

Sets the MPIDR.MT bit. Setting this to true hints the the cluster is multi-threading compatible.

**swp\_with\_xzr\_is\_st\_atomic****Type**

bool

**Default value**

0x1

**Description**

If true, swp with dest as xzr is treated as store atomic.

**take\_ccfail\_tsc\_trap****Type**

bool

**Default value**

0x0

**Description**

When take\_ccfail\_undef=1 this parameter controls whether or not an SMC instruction that is trapped by HCR\_EL2.TSC but fails its condition code check generates a trap to EL2.

**take\_ccfail\_undef****Type**

bool

**Default value**

0x1

**Description**

UNDEF exception is taken even if condition code check fails.

**tidcp\_traps\_el0\_undef\_imp\_def****Type**

bool

**Default value**

0x1

**Description**

TIDCP has priority over UNDEF for accesses to IMPLEMENTATION DEFINED functionality from EL0.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_or\_ic\_invalid\_xt****Type**

int

**Default value**

0x0

**Description**

Behavior of TLBI and IC instructions that don't take Xt as an argument when Xt != 0b11111.  
0: TLB and IC not UNDEF, 1: TLBI UNDEF, IC not UNDEF, 2: TLBI not UNDEF, IC UNDEF, 3: TLBI and IC UNDEF.

**trace\_has\_sysreg\_access****Type**

bool

**Default value**

0x1

**Description**

ETM trace registers support access via system registers.

**trace\_icc\_registers\_as\_icv\_when\_redirected****Type**

bool

**Default value**

0x0

**Description**

If true, update trace with ICV, instead of ICC when ICV registers are accessed depending on the core state.

**trace\_physical\_registers\_when\_host\_virtualisation\_enabled****Type**

int

**Default value**

0x0

**Description**

When host virtualisation is enabled, trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers).

**trace\_xzr\_in\_core\_regs64\_trace****Type**

bool

**Default value**

0x1

**Description**

Whether CORE\_REGS64\_READ traces XZR and WZR input registers.

**trap\_dc\_cmo\_to\_pou\_if\_nop**

**Type**

bool

**Default value**

0x1

**Description**

Whether traps to DC CMO operations to PoU are ignored if the same is treated as NOP.

**trap\_ic\_cmo\_to\_pou\_if\_nop**

**Type**

bool

**Default value**

0x1

**Description**

Whether traps to IC CMO operations to PoU are ignored if the same is treated as NOP.

**trap\_reserved\_group3\_id\_regs**

**Type**

bool

**Default value**

0x0

**Description**

Whether setting HCR\_EL2.TID3 traps reserved group3 id registers.

**treat-dcache-cmos-to-poc-as-nop**

**Type**

int

**Default value**

0x0

**Description**

Whether dcache maintenance operations to the point of coherency are required for instruction to data coherence. 0 - Clean/Invalidate ops required, 1 - Clean/Invalidate ops not required and cannot generate faults, 2 - Clean/Invalidate ops not required but can generate faults.

**treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**treat-dcache-invalidate-as-clean-invalidate****Type**

bool

**Default value**

0x0

**Description**

Treat data cache invalidate operations as clean and invalidate.

**treat-icache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

If has\_coherent\_icache is true, whether instruction cache invalidation operations to PoU which are treated as NOP can generate fault. 0 - cannot generate faults, 1 - can generate faults.

**treat\_forced\_normal\_as\_device\_for\_excl\_atomics****Type**

bool

**Default value**

0x0

**Description**

Whether exclusive/atomic access is supported in same manner as access to device if stage1 is Device memory and final memory attribute forced to normal by FWB.

**treat\_pld\_as\_nop****Type**

bool

**Default value**

0x0

**Description**

If true, treat PLD as NOP.

**`treat_pli_as_nop`****Type**

bool

**Default value**

0x0

**Description**

If true, treat PLI as NOP.

**`treat_wfi_wfe_as_nop`****Type**

bool

**Default value**

0x0

**Description**

If true, never go into wait state for WFI or WFE instructions.

**`truncate_pc_on_illegal_exception_return_to_aarch32`****Type**

bool

**Default value**

0x1

**Description**

On Illegal ERET to AArch32, truncate PC to 32-bits.

**`unification-level`****Type**

int

**Default value**

0x1

**Description**

Level of Unification Inner Shareable for the cache hierarchy.

**`unification-uniprocessor-level`****Type**

int



**Default value**

0x1

**Description**

Level of Unification Uniprocessor for the cache hierarchy.

**unpred\_LSE128\_overlap****Type**

int

**Default value**

0x1

**Description**

Constrained unpredictable behaviours for 128-bit LSE overlap. 1 Constraint\_UNDEF, 2 Constraint\_NOP.

**unpred\_brbe\_next\_branch\_cycle\_count\_unknown****Type**

bool

**Default value**

0x0

**Description**

If true, cycle count value for the next BRBE branch record after BRB INJ execution outside prohibited region is unknown.

**unpred\_clear\_ISV\_for\_exception\_before\_software\_step****Type**

bool

**Default value**

0x0

**Description**

Whether ESR\_ELx.ISV bit is cleared/set, when it is constrained unpredictable due to a different exception before a software step exception.

**unpred\_edscr\_ns\_set\_unknown\_bit****Type**

bool

**Default value**

0x0

**Description**

Unknown(x) bit in NS field in EDSCR can be configure to 0 or 1.

**unpred\_edscr\_rw\_unknown\_bits\_read\_as\_1****Type**

bool

**Default value**

0x0

**Description**

Unknown(x) bits in RW field in EDSCR are read as 1 instead of 0.

**unpred\_edscr\_status\_read\_as\_no\_syndrome****Type**

bool

**Default value**

0x0

**Description**

Controls the choice of EDSCR.STATUS bit-values, when it is constrained unpredictable behaviour due to a different exception before a halting step debug event.

**unpred\_extdbg\_unknown\_bits****Type**

int

**Default value**

0x0

**Description**

Data used to fill only in UNKNOWN bit-fields of external debug registers e.g., EDPFR and EDDFR.

**unpred\_load\_single\_reg\_overlap\_with\_wb****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable behaviours for single load with writeback(might impact certain load pair instructions) 0 Constraint\_WBSUPPRESS, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

**unpred\_mrsmsr\_currentlymapped\_undef****Type**

bool

**Default value**

0x0

**Description**

UNPREDICTABLE register access (accessible from current mode using different instruction) modeled as NOP when false and UNDEF when true.

**unpred\_mrsmsr\_protfailed\_undef****Type**

bool

**Default value**

0x0

**Description**

UNPREDICTABLE register access (not accessible from current PL and security state) modeled as NOP when false and UNDEF when true.

**unpred\_par\_attr\_returns\_mair****Type**

bool

**Default value**

0x0

**Description**

If true, PAR\_EL1.ATTR represents the memory attributes as per the MAIR value instead of the ones in the descriptor.

**unpred\_sctlr\_c\_0\_taggable\_behaviour****Type**

int

**Default value**

0x2

**Description**

Controls unpredictable effects when SCTLTR\_ELx.C=0 for a stage 1 translation regime on whether memory is treated as taggable. Values: 0=Tagged, 1=Untagged but forced to Tagged when FWB=1 and stage 2 restores WB, 2 = Untagged.

**unpred\_stage2\_mpu\_and\_bg\_disabled****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable when stage2 MPU and background disabled. 0, Stage-2 level 0 translation fault(Default). 1, Unknown memory attributes.

**unpred\_store\_exclusive\_base\_overlap****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable behaviours for store exclusive when s==n. 0 Constraint\_NONE, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

**unpred\_store\_pair\_and\_single\_reg\_overlap\_with\_wb****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable behaviours for pair and single store with writeback(doesn't cover store exclusive) 0 Constraint\_NONE, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

**unpred\_tlbi\_not\_in\_monitor\_mode****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable behaviors for AArch32 TLBI instructions executed in secure privileged mode other than Monitor mode. 0: Preferred behavior (default), 1: UNDEF, 2: NOP, 3: execute as if had been executed in Monitor mode.

**unpred\_tsize\_aborts****Type**

bool

**Default value**

0x0

**Description**

Behaviour when TSize is out of range. 0, force into range. 1, translation fault, forces unpred\_tsize\_pamax\_aborts to 1.

**unpred\_tsize\_pamax\_aborts****Type**

bool

**Default value**

0x0

**Description**

Behaviour when stage 2 TSize exceeds the physical address size (or 40bits, from AArch32). 0, force into range. 1, translation fault. Ignored if unpred\_tsize\_aborts is 1.

**unpredictable\_exclusive\_abort\_memtype****Type**

int

**Default value**

0x0

**Description**

Cause MMU abort if exclusive access is not supported in certain memory type (0=exclusives allowed in all memory types, 1=exclusives abort in Device memory types, 2=exclusives abort in any type other than WB inner cacheable).

**unpredictable\_hvc\_behaviour****Type**

int

**Default value**

0x0

**Description**

HVC unpredictable behaviour. 0, UNDEF. 1, NOP.

**unpredictable\_smc\_behaviour****Type**

int

**Default value**

0x0

**Description**

SMC unpredictable behaviour. 0, UNDEF. 1, NOP.

**unpredictable\_wfet\_and\_wfit\_behaviour****Type**

int

**Default value**

0x1

**Description**

WFET and WFIT unpredictable behaviour in debug state. 0, UNDEFINED. 1, NOP.

**unsupported\_atomic\_fault\_type****Type**

int

**Default value**

0x0

**Description**

Type of fault reported on unsupported atomic access. 0 = external abort if any reported by interconnect, 1 = precise unsupported atomic fault, 2 = precise external abort, 3 = imprecise external abort.

**unsupported\_hw\_update\_fault\_type****Type**

int

**Default value**

0x0

**Description**

Type of abort reported when hw update to descriptor is done using unsupported memtype (0=No abort, 1=IMPDEF abort caused by memtype, 2=Sync external abort).

**use\_architectural\_names****Type**

bool

**Default value**

0x0

**Description**

Use names SP/LR/PC instead of R13/R14/R15.

**use\_stage1\_sh\_as\_input\_to\_stage2****Type**

bool

**Default value**

0x0

**Description**

IMPDEF case of whether to use stage1 shareability or OuterShareable as input to stage2 if stage1 is Device memtype.

**use\_tlb\_contig\_hint****Type**

bool

**Default value**

0x0

**Description**

Translation table entries with the contiguous hint bit set generate large TLB entries.

**user\_defined\_rom\_table\_debug\_power\_config****Type**

string

**Default value**

""

**Description**

User defined ROM Table debug power domains for ED,CTI,PMU and TRACE, and DBGPCR configuration. The "version" field and "cores" array are mandatory. The "dbgpcr" array, if provided, must contain unique integers in the range [0, 32) describing which debug power domains have power control implemented. The "rom" and "dbgpcr" fields in objects in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. All power domain ID fields ("rom", "ed/pmu", "cti", "etm") must be in the range [0, 32). The "ed/pmu" field is mandatory. Example JSON for a hierarchical debug ROM layout: {"version": 0, "dbgpcr": [0, 1], "cores": [{"dbgpcr": [1, 31], "rom": 0, "ed/pmu": 0, "cti": 31, "etm": 1}, {"ed/pmu": 0}]}

**vpu\_datapath\_width****Type**

int

**Default value**

0x80

**Description**

VPU data path width.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**warn\_unpredictable\_in\_v7****Type**

bool

**Default value**

0x1

**Description**

If true, behaviour which is unpredictable in V7 yet is predictable in V8 will produce a warning.

**watchpoint-log2secondary\_restriction****Type**

int

**Default value**

0x0

**Description**

log2 size of secondary restriction of FAR/EDWAR possible values on watchpoint hit for load/store operations.

**wfe\_wakeup\_delay****Type**

int

**Default value**

0x0

**Description**

Configure WFE wakeup delay in CPU cycles.

**wfi\_wakeup\_delay****Type**

int

**Default value**

0x0

**Description**

Configure WFI wakeup delay in CPU cycles.

**wp\_ignores\_dbm\_update****Type**

bool

**Default value**

0x0

**Description**

If true, dbm update is ignored on watchpoint hit.



### 3.5.2 AEMvACT

ARM AEM A-Profile(MP) CPU component - number of cores configurable at runtime. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-186: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Changes in 11.29.19

Parameters added:

- `atomic_memtype_fault_prio_less_than_gpc_fault`
- `brbe_disable_recording`
- `e2h_forces_interrupt_overrides`
- `enable-gicv5`
- `gcspr_sync_immediate`
- `gicv5.config_file`
- `gicv5.interrupt-bypass-support`
- `has_mbist_never1_ae25`
- `par_nse_set_unknown_bit`
- `pmbsr_reports_external_abort`
- `sve.smidr_el1_nsmc_val`
- `sve.smidr_el1_sh_val`
- `tune_spe_cache_events`
- `unpred_brbe_next_branch_cycle_count_unknown`
- `warn_for_dbgwcr_reserved_values_with_razwi_bits`

Parameters removed:

- `use_Xt_as_LDG_STG_input`
- `use_mte_eac_02_instructions_encoding`
- `use_mte_eac_08_tfsr_encoding`

#### About AEMvACT

AEMvACT implements all architectural features in Arm®v8-A and Armv9-A.

It provides parameters to enable or disable support for particular architectural features. Some of these parameters allow you to enable features in versions of the architecture earlier than the one in which they were introduced. They have the possible values:

- 0** Feature is not enabled.
- 1** Feature is enabled, but only if the model is set to implement a version of the architecture in which the feature is supported.
- 2** Feature is enabled, regardless of which architecture version the model implements.

This model supports the Arm®v8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Mapping architectural features to Fast Models releases

The following table shows which Fast Models releases support new architectural features, from Armv8.3-A onwards. Support is at EAC quality unless otherwise stated.

**Table 3-187: Architectural features implemented in Fast Models**

Architecture version	Feature	Fast Models version									
		11.29	11.28	11.27	11.26	11.25	11.24	11.23	11.22	11.21	11.20
Armv8.3-A	All. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Armv8.4-A	All. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Armv8.5-A	Memory Tagging Extension (MTE) functionality	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	MTE performance	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	All except MTE. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Armv8.6-A	All. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Armv8.7-A	Limited TLBI maintenance	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	All except TLBI maintenance. For details, see list after table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Armv8.8-A	All	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Armv8.9-A	Guarded Call Stack, 128-bit page table descriptors, and PMU updates	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	All other features	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Architecture version	Feature	Fast Models version									
		11.29	11.28	11.27	11.26	11.25	11.24	11.23	11.22	11.21	11.20
Armv9.0-A to Armv9.3-A	Branch Record Buffer Extension (BRBE).	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Trace Buffer Extension (TRBE) and Embedded Trace Extension (ETE). Reading TRBE registers was supported from 11.15.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Realm Management Extension (RME) and Scalable Matrix Extension (SME).	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Scalable Matrix Extension 2 (SME2).	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	All other features. For details, see the list below this table.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Armv9.4-A	RME without Secure EL2.	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	All other features.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Armv9.5-A	All features.	✓	✓	✓	✓	✓	✓	-	-	-	-
Armv9.6-A	All features.	✓	✓	-	-	-	-	-	-	-	-

- Armv8.3-A features implemented include:
  - Pointer authentication
  - Nested virtualization
  - Advanced SIMD complex number support
  - Improved Javascript data type conversion support
  - Memory consistency model changes
  - Support for larger system-visible caches
  - Debug over powerdown
- Armv8.4-A features implemented include:
  - Secure EL2
  - Additional cryptographic hashes
  - Activity monitors
  - Enhanced support for nested virtualization

- Memory Partitioning and Monitoring (MPAM)
- Stage 2 forced write-back
- Dot product instructions
- Data independent timing instructions
- Large System Extensions (LSE)
- Support for TLB maintenance instructions and for TLB range instructions
- Translation Table Level (TTL)
- Enhancements to weaker release consistency
- Debug relaxations and extensions
- Flag manipulation instructions
- Armv8.5-A features implemented include:
  - Branch Target Indicators (BTI)
  - Enhanced Virtualization Traps (EVT)
  - Random Number Generator (RNG)
  - Flag Manipulation instructions
  - Floating-point round to integer
  - DC CVADP
- Armv8.6-A features implemented include:
  - General Matrix Multiply (GEMM)
  - Enhanced PAC2
  - FPAC
  - Data Gathering Hint
  - Additional PMU events
  - Enhanced Counter Virtualization
- Armv8.7-A features implemented include:
  - WFE and WFI with timeouts
  - Larger physical address for 4KB and 16KB translation granules
  - MTE Asymmetric Fault Handling
  - Enhancements to Privilege Access Never (PAN)
  - Enhancements to PMU and SPE
- Other Armv9-A features implemented include:
  - Scalable Vector Extension version 2 (SVE2)
  - Transactional Memory Extension (TME)

## AEMvA core personalities

The AEM core personalities feature allows you to configure AEM instances in a platform to use the **IMPLEMENTATION\_DEFINED** registers and **UNPREDICTABLE** behavior for a specific implementation of your choice at model startup.

Configuring an AEM with a core personality requires a license for both the AEM and the selected implementation.

Set the personality using either:

- The environment variable `FASTSIM_AEM_A_PROFILE`.
- The parameter `impdef_regs_and_unpred_from_implementation`.

The parameter allows you to configure different instances in the same platform with different personalities, including subclusters in a heterogeneous AEM. The environment variable takes precedence over the parameter and affects all instances of the AEM in the platform. Otherwise the two options function identically.

To see the available values for the environment variable or parameter, set either of them to the special value `list`. The model prints the list of available values and exits. An example value is `ARM_Cortex-X2`. Then set the parameter or environment variable to the required value.

Configuring a core personality only affects **IMPLEMENTATION\_DEFINED** registers and **UNPREDICTABLE** behavior. In other respects, the cluster or subcluster still behaves like the AEM. For example, all parameters default to the AEM default values. Therefore, you must manually set parameters to valid values for the configured personality, as many of the defaults are incompatible with any given implementation.

To assist with this configuration, the AEM prints warnings for any parameter with an invalid value for the configured personality. The warning lists the parameter name and the valid value or range of values it can be set to for the selected personality.



Running the model with invalid parameter configurations for the selected personality can lead to unexpected behavior.

## Iris and MTI instances for AEMvACT

This model has the following Iris instances:

**Table 3-188: AEMvACT Iris instances**

InstanceName	ComponentName
AEMvACT	Cluster_ARM_AEM-A_MP
AEMvACT.AMU	PVBusLogger
AEMvACT.AMU.mapper	PVBusMapper
AEMvACT.DAP	PVBusLogger

InstanceName	ComponentName
AEMvACT.DAP.mapper	PVBusMapper
AEMvACT.DSU	DSU
AEMvACT.DSU.mpam_busslave	PVBusSlave
AEMvACT.MMAP	PVBusLogger
AEMvACT.MMAP.mapper	PVBusMapper
AEMvACT.RAS	PVBusLogger
AEMvACT.RAS.mapper	PVBusMapper
AEMvACT.acp_mapper	PVBusMapper
AEMvACT.cpu0	ARM_AEM-A_MP
AEMvACT.cpu0.UTLB	TLB
AEMvACT.cpu0.debug_rom	debug_rom
AEMvACT.cpu0.dtlb	TLB
AEMvACT.cpu0.l1dcache	PVCache
AEMvACT.cpu0.l1dcache.upstream[0]	PVBusSlave
AEMvACT.cpu0.l1icache	PVCache
AEMvACT.cpu0.l1icache.upstream[0]	PVBusSlave
AEMvACT.ext_bus	PVBusLogger
AEMvACT.ext_bus.mapper	PVBusMapper
AEMvACT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
AEMvACT.global_debug_rom	debug_rom
AEMvACT.l2_cache	PVCache
AEMvACT.l2_cache.upstream[0]	PVBusSlave
AEMvACT.l2_cache.upstream[10]	PVBusSlave
AEMvACT.l2_cache.upstream[11]	PVBusSlave
AEMvACT.l2_cache.upstream[12]	PVBusSlave
AEMvACT.l2_cache.upstream[13]	PVBusSlave
AEMvACT.l2_cache.upstream[14]	PVBusSlave
AEMvACT.l2_cache.upstream[15]	PVBusSlave
AEMvACT.l2_cache.upstream[16]	PVBusSlave
AEMvACT.l2_cache.upstream[1]	PVBusSlave
AEMvACT.l2_cache.upstream[2]	PVBusSlave
AEMvACT.l2_cache.upstream[3]	PVBusSlave
AEMvACT.l2_cache.upstream[4]	PVBusSlave
AEMvACT.l2_cache.upstream[5]	PVBusSlave
AEMvACT.l2_cache.upstream[6]	PVBusSlave
AEMvACT.l2_cache.upstream[7]	PVBusSlave
AEMvACT.l2_cache.upstream[8]	PVBusSlave
AEMvACT.l2_cache.upstream[9]	PVBusSlave
AEMvACT.l2_flusher	AsyncCacheFlushUnit

InstanceName	ComponentName
AEMvACT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-189: AEMvACT MTI instances**

InstanceName	ComponentName
AEMvACT.AMU	PVBusLogger
AEMvACT.AMU.mapper	PVBusMapper
AEMvACT.DAP	PVBusLogger
AEMvACT.DAP.mapper	PVBusMapper
AEMvACT.DSU	DSU
AEMvACT.DSU.mpam_busslave	PVBusSlave
AEMvACT.MMAP	PVBusLogger
AEMvACT.MMAP.mapper	PVBusMapper
AEMvACT.RAS	PVBusLogger
AEMvACT.RAS.mapper	PVBusMapper
AEMvACT.acp_mapper	PVBusMapper
AEMvACT.cpu0	ARM_AEM-A_MP
AEMvACT.cpu0.UTLB	TLB
AEMvACT.cpu0.l1dcache	PVCache
AEMvACT.cpu0.l1dcache.upstream[0]	PVBusSlave
AEMvACT.cpu0.l1icache	PVCache
AEMvACT.cpu0.l1icache.upstream[0]	PVBusSlave
AEMvACT.ext_bus	PVBusLogger
AEMvACT.ext_bus.mapper	PVBusMapper
AEMvACT.gic_cpuif_decoder_cluster	GlCv3CPUInterfaceDecoder
AEMvACT.l2_cache	PVCache
AEMvACT.l2_cache.upstream[0]	PVBusSlave
AEMvACT.l2_cache.upstream[10]	PVBusSlave
AEMvACT.l2_cache.upstream[11]	PVBusSlave
AEMvACT.l2_cache.upstream[12]	PVBusSlave
AEMvACT.l2_cache.upstream[13]	PVBusSlave
AEMvACT.l2_cache.upstream[14]	PVBusSlave
AEMvACT.l2_cache.upstream[15]	PVBusSlave
AEMvACT.l2_cache.upstream[16]	PVBusSlave
AEMvACT.l2_cache.upstream[1]	PVBusSlave
AEMvACT.l2_cache.upstream[2]	PVBusSlave
AEMvACT.l2_cache.upstream[3]	PVBusSlave
AEMvACT.l2_cache.upstream[4]	PVBusSlave
AEMvACT.l2_cache.upstream[5]	PVBusSlave

InstanceName	ComponentName
AEMvACT.l2_cache.upstream[6]	PVBusSlave
AEMvACT.l2_cache.upstream[7]	PVBusSlave
AEMvACT.l2_cache.upstream[8]	PVBusSlave
AEMvACT.l2_cache.upstream[9]	PVBusSlave
AEMvACT.l2_flusher	AsyncCacheFlushUnit

## Ports for AEMvACT

**Table 3-190: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcasttinner	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
cfgend[8]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
CNTHPIRQ[8]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[8]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[8]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[8]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[8]	Signal	Master	Timer signals to SOC.



Name	Protocol	Type	Description
commirq[8]	Signal	Master	Interrupt signal from debug communications channel.
config64[8]	Signal	Slave	Register width after reset.
cp15sdisable[8]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[8]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
CRITICALIRQ[8]	Signal	Master	RAS Critical Error Interrupt.
cryptodisable[8]	Signal	Slave	Disable cryptography extensions after reset.
cti[8]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
cti0extin[4]	Signal	Slave	CTI trace inputs for core 0.
cti0extout[4]	Signal	Master	CTI trace outputs for core 0.
cti1extin[4]	Signal	Slave	CTI trace inputs for core 1.
cti1extout[4]	Signal	Master	CTI trace outputs for core 1.
cti2extin[4]	Signal	Slave	CTI trace inputs for core 2.
cti2extout[4]	Signal	Master	CTI trace outputs for core 2.
cti3extin[4]	Signal	Slave	CTI trace inputs for core 3.
cti3extout[4]	Signal	Master	CTI trace outputs for core 3.
cti4extin[4]	Signal	Slave	CTI trace inputs for core 4.
cti4extout[4]	Signal	Master	CTI trace outputs for core 4.
cti5extin[4]	Signal	Slave	CTI trace inputs for core 5.
cti5extout[4]	Signal	Master	CTI trace outputs for core 5.
cti6extin[4]	Signal	Slave	CTI trace inputs for core 6.
cti6extout[4]	Signal	Master	CTI trace outputs for core 6.
cti7extin[4]	Signal	Slave	CTI trace inputs for core 7.
cti7extout[4]	Signal	Master	CTI trace outputs for core 7.
ctidbgirq[8]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen[8]	Signal	Slave	External debug interface.
dbgnopwrdown[8]	Signal	Master	Debug no power down request.
dbgpwrdownack[8]	Signal	Master	Debug power down acknowledge.
dbgpwrdownreq[8]	Signal	Slave	Debug power down request.
dbgpwrupreq[8]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
ERRORIRQ[8]	Signal	Master	RAS Error Recovery Interrupt.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
external_trace_reset[8]	Signal	Slave	ETMv4 External Trace Reset signal.
FAULTIRQ[8]	Signal	Master	RAS Fault Handling Interrupt

Name	Protocol	Type	Description
fiq[8]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
fiq_nmi[8]	Signal	Slave	-
gicv3_redistributor_s[8]	GICv3Comms	Slave	GICv3 cpu interface ports.
hacdbsirq[8]	Signal	Master	Interrupt signal from the HACDBS unit.
irq[8]	Signal	Slave	This signal drives the CPUs interrupt handling.
irq_nmi[8]	Signal	Slave	-
irqs[224]	Signal	Slave	These signals drive the CPU's interrupt controller interrupt lines.
l2reset	Signal	Slave	This signal resets timer and interrupt controller.
memorymapped_amu_s	PVBus	Slave	External interface for amu.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[8]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.
pmbirq[8]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[8]	Signal	Master	Interrupt signal from performance monitoring unit.
pmusnapshotacks[8]	Signal	Master	-
pmusnapshotreqs[8]	Signal	Slave	-
presetdbg[8]	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[8]	Signal	Slave	Individual processor RAM Error Interrupt signal input.
reset[8]	Signal	Slave	Raising this signal will put the core into reset mode.
rlpiden[8]	Signal	Slave	External debug interface.
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
rtpiden[8]	Signal	Slave	External debug interface.
rvbar[8]	Value_64	Slave	Reset vector base address.
sei[8]	Signal	Slave	Per core System Error physical pins.
smpnamp[8]	Signal	Master	This signals AMP or SMP mode for each core.
spiden[8]	Signal	Slave	External debug interface.
spniden[8]	Signal	Slave	External debug interface.
standbywfe[8]	Signal	Master	This signal indicates if a core is in WFE state.

Name	Protocol	Type	Description
standbywfi[8]	Signal	Master	This signal indicates if a core is in WFI state.
standbywfil2	Signal	Master	This signal indicated all cores and L2 are in a power down state
teinit[8]	Signal	Slave	This signal provides default exception handling state.
ticks[8]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trace_unit_reset[8]	Signal	Slave	ETMv4 Trace Unit Reset signal.
trbirq[8]	Signal	Master	Interrupt signal from the trace buffer unit.
vcpumntirq[8]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[8]	Signal	Slave	Virtual FIQ input. Note that the irq/fiq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
vinithi[8]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[8]	Signal	Slave	Virtual IRQ input. Note that the irq/fiq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[8]	Signal	Slave	Processor Virtual System Error Interrupt request.

## Parameters for AEMvACT

### ADFSR-AIFSR-implemented

#### Type

bool

#### Default value

0x0

#### Description

ADFSR and AIFSR are implemented.

### AIDR

#### Type

int

#### Default value

0x0

**Description**

Value of AIDR\_EL1 register.

**AMIIDR****Type**

int

**Default value**

0x43b

**Description**

Value of AMU Implementation Identification Register.

**AMPIDR****Type**

int

**Default value**

0x4000bb000

**Description**

Value of AMU Peripheral Identification Register.

**BPIMVA\_causes\_translation\_lookup****Type**

bool

**Default value**

0x0

**Description**

Do a translation when BPIMVA instruction is executed (which may cause a translation fault).

**BROADCASTCACHEMAIN****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTINNER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**CCSIDR-L1D\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L1I\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L2\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L3\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, allow L3 selection in CSSELR and present this value in CCSIDR (this is cosmetic and does not affect cache behaviour).

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**CTIPIDR****Type**

int

**Default value**

0x0

**Description**

If non-zero, override the CTI Peripheral Identification Register.

**CTR-L1Ip-override****Type**

int

**Default value**

0x0

**Description**

If non-zero, override the L1Ip bits in CTR/CTR\_ELO system register. This does not change the behaviour of the cache, only what is present in the CTR register.

**DBGBCR\_BT\_applies\_RES0\_before\_valid\_check****Type**

bool

**Default value**

0x1

**Description**

If true, RES0 behaviour is applied to DBGBCR(\_EL1).BT before checking for reserved values for this field.

**DBGPIDR****Type**

int

**Default value**

0x0

**Description**

If non-zero, override the Debug Peripheral Identification Register.

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x0

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**ERRIIDR****Type**

int

**Default value**

0xd800143b

**Description**

Value of RAS Implementation Identification Register.

**ERRPIDR****Type**

int

**Default value**

0x4100bbd80

**Description**

Value of RAS Peripheral Identification Register.

**ERXMISCO\_mask****Type**

int



**Default value**

0x0

**Description**

Write Mask for ERXMISC0 RAS Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**GMID-log2-block-size****Type**

int

**Default value**

0x4

**Description**

Log2 of the block size accessed by STGM/LDGM/STZGM instructions.

**ISV\_set\_to\_0\_for\_stage2\_synch\_external\_abort****Type**

bool

**Default value**

0x0

**Description**

Whether ESR\_EL2.ISV is set to 0 on stage 2 synchronous external aborts.

**MIDR****Type**

int

**Default value**

0x410fd0f0

**Description**

Value of MIDR\_EL1 register.

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**PA\_SIZE****Type**

int

**Default value**

0x28

**Description**

Physical address range supported For ARMv8.0 and ARMv8.1 this is limited to 48 bits (FEAT\_LPA).

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**PMCEID0****Type**

int

**Default value**

0xffffffff

**Description**

Performance Monitor Common Event ID Reg 0 value - 64 bit.

**PMCEID1****Type**

int

**Default value**

0xffffffff

**Description**

Performance Monitor Common Event ID Reg 1 value - 64 bit.

**PMSIDR.ArchInst****Type**

bool

**Default value**

0x1

**Description**

Defines whether architecture instruction sampling is implemented or not, if not only micro op sampling is implemented. Model only supports architecture instruction sampling, but allows ID register field to be configured.

**PMSIDR.CRR****Type**

bool

**Default value**

0x0

**Description**

Defines whether call return branch records (FEAT\_SPE\_CRR) is implemented or not.

**PMSIDR.LDS****Type**

bool

**Default value**

0x0

**Description**

Defines whether data source for sampled load instruction is implemented or not. Model does not implement loaded data source, but allows ID register field to be configured.

**PMUPIDR****Type**

int

**Default value**

0x0

**Description**

If non-zero, override the PMU Peripheral Identification Register.

**abort\_execution\_from\_device\_memory****Type**

bool

**Default value**

0x0

**Description**

Execution from device memory generates a prefetch abort.

**advsimd\_bf16\_support\_level****Type**

int

**Default value**

0x0

**Description**

Implement BFloat16 operations from ARMv8.6. AArch64 Advanced SIMD and FP BFloat16 instructions are automatically enabled when has\_arm\_v8-6 is true. - 0, Not implemented. - 1, AArch64 Advanced SIMD and FP BFloat16 instructions only (FEAT\_BF16). - 2, AArch32 Advanced SIMD and VFP BFloat16 instructions only (FEAT\_AA32BF16). - 3, Both AArch64 Advanced SIMD and FP and AArch32 Advanced SIMD and VFP BFloat16 instructions.

**advsimd\_i8mm\_support\_level****Type**

int

**Default value**

0x0

**Description**

Implement Int8 matrix multiply operations from ARMv8.6. AArch64 Advanced SIMD and FP Int8 matrix multiply instructions are automatically enabled when has\_arm\_v8-6 is true. - 0, Not implemented. - 1, AArch64 Advanced SIMD and FP Int8 matrix multiply instructions only (FEAT\_I8MM). - 2, AArch32 Advanced SIMD and VFP Int8 matrix multiply instructions only (FEAT\_AA32I8MM). - 3, Both AArch64 Advanced SIMD and FP and AArch32 Advanced SIMD and VFP Int8 matrix multiply instructions (FEAT\_I8MM, FEAT\_AA32I8MM).

**advsimd\_overread****Type**

bool

**Default value**

0x0

**Description**

AdvSIMD element load operations access all bytes of a 16-byte aligned window, even in Device memory.

**align\_pc\_on\_branch\_to\_unaligned\_pc\_aarch32****Type**

bool

**Default value**

0x0

**Description**

Force PC align for branches to an unaligned PC counter in A32 state.

**align\_pc\_on\_debug\_exit\_to\_aarch32****Type**

bool

**Default value**

0x0

**Description**

Exit to AARCH32 state from debug state forces pc bit0 to 0.

**align\_pc\_on\_illegal\_exception\_return\_to\_aarch32****Type**

bool

**Default value**

0x1

**Description**

Align PC when performing an illegal exception return from AArch64 to AArch32.

**allow\_s1\_dbm\_update\_on\_s2\_mmu\_fault****Type**

bool

**Default value**

0x1

**Description**

Whether s1 dirty bit update is done when s2 of ipa (not s1 ttw) generates mmu fault.

**amair\_reg\_rw\_mask****Type**

int

**Default value**

0x0

**Description**

RW mask for implementation-defined registers.

**amu\_aux\_type\_fixed****Type**

string

**Default value**

""

**Description**

Lists which AMU auxiliary registers that are fixed and to which event type. The JSON schema is: {fixed\_aux\_reg:evt\_type, ...}. For example {"0":0x300} would make auxiliary register 0 fixed to event type 0x300.

**amu\_aux\_voffset\_mask****Type**

int

**Default value**

0x0

**Description**

If ARMv8.6 is implemented, each bit of the field, 0 to 15, when 1 indicates that the corresponding virtual offset register, AMEVCNTVOFF1<n>\_EL2, is implemented.

**amu\_has\_external\_interface****Type**

int

**Default value**

0x0

**Description**

Implement external memory-mapped access to system register of activity monitor unit from ARMv8.4 (FEAT\_AMU\_EXT). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**amu\_has\_sysreg\_interface****Type**

bool

**Default value**

0x1

**Description**

Implement system register access to activity monitor unit from ARMv8.4.

**amu\_mmap\_address****Type**

string

**Default value**

""

**Description**

AMU base address for each core on system bus. 0 means the AMU is not mapped, otherwise the address must be 4KB aligned. JSON schema for the parameter value is: {"format": "all\_addrs\_are\_absolute\_wrt\_systembus", "cores": [{"amu": 0x0}, {"amu": 0x0}, {"amu": 0x0}, {"amu": 0x0}]}.

**amu\_num\_auxiliary\_counters****Type**

int

**Default value**

0x0

**Description**

Number of AMU auxiliary counters implemented.

**amu\_reset\_domain****Type**

int

**Default value**

0x0

**Description**

Reset domain for activity monitor unit. 0, COLD\_RESET. 1, WARM\_RESET. 2, NONE.

**amu\_version****Type**

int

**Default value**

0x1

**Description**

Selects the activity monitor version implemented - 1, AMUv1 for Armv8.4 is implemented. - 2, AMUv1 for Armv8.6 is implemented (FEAT\_AMUv1p1).

**apsr\_read\_restrict****Type**

bool

**Default value**

0x0

**Description**

At EL0, unknown bits of APSR are RAZ.

**arm\_v8\_7\_accelerator\_support\_level****Type**

int

**Default value**

0x0

**Description**

Implements accelerator support instructions: 0, Not implemented 1, FEAT\_LS64 implemented 2, FEAT\_LS64\_V implemented 3, FEAT\_LS64\_ACCDATA implemented 4, FEAT\_LS64WB implemented.

**atomic\_memtype\_fault\_prio\_less\_than\_gpc\_fault****Type**

bool

**Default value**

0x0

**Description**

If true, unsupported atomic/exclusive memtype faults are lower priority than GPC faults.

**atomic\_memtype\_fault\_priority****Type**

int

**Default value**

0x0

**Description**

This parameter describes the priority of unsupported atomic/exclusive memtype fault w.r.t alignment and permission fault. 0, BEFORE\_ALIGN\_MEM\_FAULT. 1, AFTER\_ALIGN\_BEFORE\_PERM\_FAULT. 2, AFTER\_PERM\_FAULT.

**auxilliary\_feature\_register0****Type**

int

**Default value**

0x0

**Description**

Value of AFR0 ID register.

**branch-predictor-clear-policy****Type**

int



**Default value**

0x2

**Description**

Set branch prediction policy as defined for MMFR1[31:28]. This does not change the behaviour of the branch predictor, only what is reported in MMFR1.BPred.

**branch-predictor-supported-ops****Type**

int

**Default value**

0x1

**Description**

Set branch prediction policy as defined for MMFR3[11:8]. This does not change the behaviour of the branch predictor, only what is reported in MMFR3.BPMaint.

**brbe\_disable\_recording****Type**

bool

**Default value**

0x0

**Description**

If BRBE is implemented and this is set to true, disable BRBE recording. All registers will be functional, but no branches will be recorded. This will improve model performance for workloads that enable BRBE, but don't care about the information stored in it. (FEAT\_BRBE).

**brbe\_log2\_num\_records****Type**

int

**Default value**

0x6

**Description**

Log2 of number of BRB records supported. 3 -> 8 records, ... 6 -> 64 records.

**brbinf\_type\_override\_on\_impdef\_trap\_to\_el3****Type**

bool

**Default value**

0x0

**Description**

If true, force BRBINF.TYPE=0x23 (trap) when ESR.EC=0x1f (implementation defined exception to EL3) (FEAT\_BRBE).

**cache-log2linelen****Type**

int

**Default value**

0x6

**Description**

Log2 of the cache line length in bytes.

**cache\_maintenance\_hits\_watchpoints****Type**

bool

**Default value**

0x0

**Description**

DCIMVA operations executed in AArch32 modes hit watchpoints.

**changing\_block\_size\_without\_bbm\_support****Type**

int

**Default value**

0x0

**Description**

Level of support for changing block size without break-before-make (FEAT\_BBM).

**check\_memory\_attributes****Type**

bool

**Default value**

0x0

**Description**

Detect and report TLB use of conflicting memory attributes for views of the same physical address.

**checked\_pointer\_arithmetic\_support\_level****Type**

int

**Default value**

0x0

**Description**

Specify the Checked Pointer Arithmetic support level: 0, not implemented. 1, FEAT\_CPA is implemented. 2, FEAT\_CPA2 is implemented.

**`clean_invalidate_cache_on_warm_reset`****Type**

bool

**Default value**

0x0

**Description**

Clean and invalidate caches on warm reset.

**`clear_IT_when_IL_set`****Type**

bool

**Default value**

0x0

**Description**

Clear IT bits when performing a \*legal\* exception return to AArch32 when IL is set.

**`clear_IT_when_IL_set_explicitly`****Type**

bool

**Default value**

0x0

**Description**

Apart from `clear_IT_when_IL_set`, also clear IT bits when loading CPSR from SPSR/memory and IL == 1 in the value being loaded.

**`clear_ec_in_debug_state`****Type**

bool

**Default value**

0x0

**Description**

When ARMv8.8 debug extension is implemented, whether EDESR.EC bit is set/cleared on entering debug state due to pending exception catch caused by EDESR.EC=1.

**clear\_reg\_top\_eret****Type**

int

**Default value**

0x1

**Description**

Behaviour of the upper 32-bits of the Xn registers when changing between AArch32 state and AArch64 state. 0, upper 32-bits preserved for all registers. 1, upper 32-bits set to 0 for all accessible registers. 2, upper 32-bits set to 0 for a random selection of accessible registers. 3, upper-32-bits set to 0 for registers touched in AArch32.

**clear\_reg\_top\_set****Type**

bool

**Default value**

0x1

**Description**

Whether to clear upper 32-bits of the Xn register when corresponding AArch32 register is set via CADI/Iris.

**configure\_pmu\_events\_with\_json****Type**

string

**Default value**

""

**Description**

"Configure v8.6 and newer PMU events. Note : This param has high priority and overrides the setting of "has\_\*\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["EVENT\_NAME\_1","EVENT\_NAME\_2"]}".

**configure\_v8\_6\_pmu\_events\_with\_json****Type**

string

**Default value**

""

**Description**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.6 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_6\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["BR\_INDNR\_RETIRE", "BR\_IND\_RETIRE",

```
"BR_RETURN_SKIP_RETIRED", "BR_RETURN_ANY_RETIRED", "BR_INDNR_SKIP_RETIRED",
"BR_INDNR_TAKEN_RETIRED", "BR_IND_SKIP_RETIRED", "BR_IND_TAKEN_RETIRED",
"BR_IMMED_SKIP_RETIRED", "BR_IMMED_TAKEN_RETIRED", "BR_SKIP_RETIRED"]}]".
```

### **configure\_v8\_8\_pmu\_events\_with\_json**

#### **Type**

string

#### **Default value**

""

#### **Description**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.8 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_8\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events": ["BR\_HINT\_COND\_RETIRED", "BR\_COND\_TAKEN\_RETIRED", "BR\_UNCOND\_RETIRED", "BR\_COND\_RETIRED", "BRNL\_TAKEN\_RETIRED", "BRNL\_IND\_TAKEN\_RETIRED", "BRNL\_INDNR\_TAKEN\_RETIRED", "BRNL\_IMMED\_TAKEN\_RETIRED", "BL\_TAKEN\_RETIRED", "BL\_IND\_TAKEN\_RETIRED", "BL\_IMMED\_TAKEN\_RETIRED"]}].

### **configure\_v8\_9\_pmu\_events\_with\_json**

#### **Type**

string

#### **Default value**

""

#### **Description**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.9 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_9\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["ASE\_SVE\_RETIRED", "ASE\_RETIRED", "VFP\_RETIRED", "SVE\_RETIRED", "CRYPTO\_RETIRED", "SIMD\_INST\_RETIRED", "ASE\_INST\_RETIRED", "SVE\_INST\_RETIRED", "ASE\_SVE\_INST\_RETIRED", "LD\_ANY\_RETIRED", "ST\_ANY\_RETIRED", "LDST\_ANY\_RETIRED", "DP\_RETIRED"]}].

### **core\_cache\_protection**

#### **Type**

int

#### **Default value**

0xffffffffffffffff

#### **Description**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

**cpacr\_trcdis\_behaviour****Type**

int

**Default value**

0x2

**Description**

Behaviour of CPACR.TRCDIS/NSACR.NSTRCDIS when there is no CP14 ETM interface. 0, RAZ/WI. 2, implemented.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CONFIG64****Type**

bool

**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.CP15SDISABLE2****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers (FEAT\_CP15SDISABLE2).

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.DCZID-log2-block-size****Type**

int

**Default value**

0x8

**Description**

Log2 of the block size cleared by DC ZVA instruction (as read from DCZID\_EL0).

**cpuX.DCZVA\_single\_write****Type**

bool

**Default value**

0x0

**Description**

Execute the DCZVA as a single write.

**cpuX.MPIDR-override****Type**

int

**Default value**

0x0

**Description**

Override of MPIDR value. If nonzero will override the MT, cluster and CPU ID bits in MPIDR.

**cpuX.RVBAR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.RVBAR32****Type**

int

**Default value**

0x0

**Description**

Reset vector address in AARCH32 when VINITHI is not set and ignore\_rvbar\_in\_aarch32 is set.

**cpuX.SMPnAMP****Type**

bool

**Default value**

0x1

**Description**

Enable broadcast messages necessary for correct SMP operation at reset.

**cpuX.TEINIT****Type**

bool



**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.aarch32\_reset\_from\_impdef\_addr****Type**

bool

**Default value**

0x1

**Description**

If PE resets into AArch32, Whether execution starts from IMPDEF address or hi/low vector.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has been built with NEON support.

**cpuX.clock\_divider****Type**

int

**Default value**

0x1

**Description**

Clock divider ratio for asymmetric MP clocking.

**cpuX.clock\_multiplier****Type**

int

**Default value**

0x1

**Description**

Clock divider ratio for asymmetric MP clocking.

**cpuX.crypto\_aes****Type**

int

**Default value**

0x2

**Description**

AES instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 2, AES and PMULL instructions implemented (FEAT\_AES, FEAT\_PMULL).

**cpuX.crypto\_sha1****Type**

int

**Default value**

0x1

**Description**

SHA-1 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA1 instructions implemented (FEAT\_SHA1).

**cpuX.crypto\_sha256****Type**

int

**Default value**

0x1

**Description**

SHA-256 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA256 instructions implemented (FEAT\_SHA256).

**cpuX.crypto\_sha3****Type**

int

**Default value**

0x0

**Description**

Implement ARMv8.4 SHA-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA3). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.crypto\_sha512****Type**

int

**Default value**

0x0

**Description**

Implement ARMv8.4 SHA-512 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA512). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.crypto\_sm3****Type**

int

**Default value**

0x0

**Description**

Implement ARMv8.4 SM-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM3). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.crypto\_sm4****Type**

int

**Default value**

0x0

**Description**

Implement ARMv8.4 SM-4 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM4). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.cti-intack\_mask****Type**

int

**Default value**

0x1

**Description**

Set bits represent that the corresponding trigger requires software acknowledge via CTIINTACK.

**cpuX.cti-number\_of\_claim\_bits****Type**

int

**Default value**

0x0

**Description**

Number of implemented bits in CTICLAIMSET.

**cpuX.cti-number\_of\_triggers****Type**

int

**Default value**

0x8

**Description**

Number of cti event triggers (default: 8, valid values: {3, 8-32}).

**cpuX.enable\_crc32****Type**

int

**Default value**

0x0

**Description**

CRC32 instructions supported. 0, not implemented. 1, CRC32 instructions implemented (FEAT\_CRC32).

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.etm-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has ETM support.

**cpuX.force-fpsid****Type**

bool

**Default value**

0x0

**Description**

Override the FPSID value.

**cpuX.force-fpsid-value****Type**

int

**Default value**

0x0

**Description**

Value to override the FPSID value to.

**cpuX.has\_hcptr\_tase****Type**

bool

**Default value**

0x1

**Description**

If false, HCPTR.TASE is RES0.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.number-of-breakpoints****Type**

int

**Default value**

0x10

**Description**

Number of breakpoints.

**cpuX.number-of-context-breakpoints****Type**

int

**Default value**

0x10

**Description**

Number of breakpoints that are context aware.

**cpuX.number-of-watchpoints****Type**

int

**Default value**

0x10

**Description**

Number of watchpoints.

**cpuX.operation\_bandwidth****Type**

int

**Default value**

0x1

**Description**

Operation width for ARMv8.4 PMU extension.

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.



**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.semihosting-stderr\_istty****Type**

bool

**Default value**

0x1

**Description**

Result for semihost istty call when argument is stderr.

**cpuX.semihosting-stdin\_istty****Type**

bool

**Default value**

0x1

**Description**

Result for semihost istty call when argument is stdin.

**cpuX.semihosting-stdout\_istty****Type**

bool

**Default value**

0x1

**Description**

Result for semihost istty call when argument is stdout.

**cpuX.semihosting-use\_stderr****Type**

bool

**Default value**

0x0

**Description**

Send stderr from the simulated process to host stderr.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.unpredictable\_WPMASKANDBAS****Type**

int

**Default value**

0x1

**Description**

Constrained unpredictable handling of watchpoints when mask and BAS fields specified. 0, IGNOREMASK. 1, IGNOREBAS (default). 2, REPEATBAS8. 3, REPEATBAS.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**cpuX.vfp-traps****Type**

bool

**Default value**

0x1

**Description**

Implement support for trapping floating-point exceptions.

**cpuX.vfp-traps-show-all****Type**

bool

**Default value**

0x0

**Description**

Report all trapped floating-point exceptions in the syndrome when a combination occurs.

**cpuX.wfet\_early\_or\_delayed\_timeout****Type**

int

**Default value**

0x0

**Description**

WFET early or delayed timeout beyond the threshold value of CNTVCT\_ELO in percentage.

**cpuX.wfit\_early\_or\_delayed\_timeout****Type**

int

**Default value**

0x0

**Description**

WFIT early or delayed timeout beyond the threshold value of CNTVCT\_ELO in percentage.

**cpuselr\_el3\_sync\_immediate****Type**

bool

**Default value**

0x1

**Description**

Adjust when the patching selection register synchronises - either immediately (true - default), or awaiting for barrier event.

**cpy\_mops\_option****Type**

int

**Default value**

0x0

**Description**

Set option for Armv8.8 CPY(FEAT\_MOPS). 0, use default(i.e. use value configured through has\_mops\_option). 1, implemented using Option A. 2, implemented using Option B.

**cpyf\_mops\_option****Type**

int

**Default value**

0x0

**Description**

Set option for Armv8.8 CPYF(FEAT\_MOPS). 0, use default(i.e. use value configured through has\_mops\_option). 1, implemented using Option A. 2, implemented using Option B.

**cycle\_counter\_freeze\_on\_spe\_event****Type**

bool

**Default value**

0x0

**Description**

If true, pmu cycle counter does not count when pmcr\_el0.dp=1 and pmu event counters are frozen by pmcr\_el0.fzs. (FEAT\_SPE\_DPFZS).

**d128\_disabled\_ps\_resvd\_size****Type**

int

**Default value**

0x2

**Description**

Physical size treated when TCR.(I)PS is programmed with value seven and D128 is disabled via TCR. 0, 48 bits. 1, 52 bits. 2, 56.

**dbg-bcr-reserved-behavior****Type**

int

**Default value**

0x1

**Description**

This is the behavior of the reserved values of the BT field in DBGBCR. Possible values are: - 0 = Disabled. - 1 = BT[2] is ignored. .

**dbg\_rom\_dap\_addr****Type**

int

**Default value**

0x0

**Description**

Debug ROM dap base address.

**dbgitr\_buffer\_size****Type**

int

**Default value**

0x0

**Description**

Number of instructions which can be buffered before EDSCR.ITE is cleared.

**dbgxvr\_ress\_is\_stateful****Type**

bool

**Default value**

0x0

**Description**

Whether DBGWVR/DBGBVR.RESS returns last written value. if set to false, RESS returns sign extended value.

**dc\_fault\_unaligned\_s1\_device\_s2\_fwb****Type**

bool

**Default value**

0x0

**Description**

Whether takes an Alignment Fault caused by the memory type on a DC {ZVA,GZVA,GVA} if the stage 1 memory type is any Device memory type.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x8

**Description**

L1 D-Cache read bus width in bytes used to calculate per-access timing annotations.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-ways****Type**

int

**Default value**

0x2

**Description**

L1 D-Cache number of ways (sets are implicit from size).

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0



**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`dcache-write_bus_width_in_bytes`****Type**

int

**Default value**

0x8

**Description**

L1 D-Cache write bus width in bytes used to calculate per-access timing annotations.

**`dcache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`dcimva_requires_s2_write_permissions`****Type**

bool

**Default value**

0x0

**Description**

Data-cache invalidate by MVA operations require stage 2 write permission (virtualised AArch32 guest).

**`dczva_reports_lowest_addr_on_tag_check_fail`****Type**

bool

**Default value**

0x0

**Description**

Whether DC ZVA reports lowest address in FAR on tag check fail.

**dczva\_wp\_far\_behaviour****Type**

int

**Default value**

0x0

**Description**

Set option for address stored in FAR/EDWARD after watchpoints hit by DC ZVA. 0 - FAR recorded matches lowest watchpointed address accessed by the instruction 1 - FAR recorded matches lowest address accessed by the instruction within same translation granule as watchpointed address 2 - FAR recorded matches highest address accessed by the instruction within same translation granule as watchpointed address.

**debug\_auth\_signals\_sampled\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Debug authentication signals can be configured as either sampled at reset only or at any time for External Root Debug.

**debug\_components\_dap\_address****Type**

string

**Default value**

""

**Description**

Debug components ROM,ED,CTI,PMU,TRACE and TRBU base address for each core on debug bus. The "rom" field in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. JSON schema for the parameter value is: {"format": "all\_addrs\_are\_absolute\_wrt\_debugbus", "cores": [{"rom": 0x0, "ed": 0x0, "cti": 0x0, "pmu": 0x0, "etm": 0x0, "trbu": 0x0}, {"rom": 0x0, "ed": 0x0, "cti": 0x0, "pmu": 0x0, "etm": 0x0, "trbu": 0x0}, {"rom": 0x0, "ed": 0x0, "cti": 0x0, "pmu": 0x0, "etm": 0x0, "trbu": 0x0}, {"rom": 0x0, "ed": 0x0, "cti": 0x0, "pmu": 0x0, "etm": 0x0, "trbu": 0x0}]}.

**debug\_components\_mmap\_address****Type**

string

**Default value**

""

## Description

Debug components ROM,ED,CTI,PMU,TRACE and TRBU base address for each core on system bus. The "rom" field in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. JSON schema for the parameter value is: {"format":"all\_addrs\_are\_absolute\_wrt\_systembus", "cores": [{"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}]}

## debug\_entry\_is\_context\_sync

### Type

bool

### Default value

0x0

### Description

If true, Entry in debug state is Context sync. Exiting debug state is a context synchronising operation, but entering is not. However some cpu implementation can consider also the Entry in Debug state as a CSE.

## debug\_rom\_is\_class\_9

### Type

bool

### Default value

0x0

### Description

If true, present a debug ROM table as a class 9 device. Otherwise, use a class 1 ROM table.

## debug\_rom\_is\_flat

### Type

bool

### Default value

0x0

### Description

If true, present a debug ROM table recommended by ARMv8 Debug Architecture. Otherwise, use nested ROM tables.

## delay\_serror

### Type

int

### Default value

0x0

**Description**

Add a propagation delay of serror signal into the core.

**delayed\_dbgreg\_between\_secure\_views****Type**

int

**Default value**

0x1

**Description**

If delayed\_dbgreg is enabled, whether the secure and nonsecure external views require explicit synchronization. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**delayed\_pmureg\_between\_secure\_views****Type**

int

**Default value**

0x1

**Description**

If delayed\_pmureg is enabled, whether the secure and nonsecure external views require explicit synchronization. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**dfr1\_reads\_actual\_bp\_wp\_ctx\_cmp****Type**

bool

**Default value**

0x0

**Description**

If true, the register ID\_AA64DFR1\_EL1/EDDFR1 reports the actual number of BRPs, WRPs and CTX\_CMPs even when the number of bp/wp/ctx\_cmp is less than 16.

**dic-spi\_count****Type**

int

**Default value**

0x40

**Description**

Number of shared peripheral interrupts implemented.

**disable\_sve\_plugin****Type**

bool

**Default value**

0x0

**Description**

If true, SVE will not be implemented in this processor even if the plugin is loaded (FEAT\_SVE).

**disable\_unknown\_update\_event\_on\_reset****Type**

bool

**Default value**

0x0

**Description**

Disables SYSREG\_UPDATE event notification on reset for the registers whose bitfields are all reserved or resets to architecturally unknown value.

**dsb\_accumulate\_threshold****Type**

int

**Default value**

0x100

**Description**

Limit the maximum number of observable DSB side effects which can be queued (e.g. TLBI), after which DSB sync will be done automatically.

**e2h\_forces\_interrupt\_overrides****Type**

bool

**Default value**

0x0

**Description**

If true, HCR\_EL2.xMO are treated as 1 else as programmed.

**early\_implicit\_error\_sync\_event\_behaviour****Type**

int

**Default value**

0x0

## Description

Set option for Early Implicit Error Synchronization event (FEAT\_IESB) 0x0 - Behavior is not described ID\_AA64MMFR4\_EL1.EIESB = 0x0 0x1 - Implicit Error synchronization event is inserted before an exception is taken to EL3 (depending on SCR\_EL3.NMEA) ID\_AA64MMFR4\_EL1.EIESB = 0x1 0x2 - Implicit Error synchronization event is inserted before an exception is taken to ELx (depending on SCR\_EL3.NMEA and SCTLR2\_ELx.NMEA) ID\_AA64MMFR4\_EL1.EIESB = 0x2 0xF - Implicit Error synchronization event is inserted after an exception is taken ID\_AA64MMFR4\_EL1.EIESB = 0xF.

## **ecv\_support\_level**

### Type

int

### Default value

0x0

### Description

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

## **eddfr1\_reads\_idreg\_mask**

### Type

int

### Default value

0xffffffffffffffff

### Description

Mask to configure each bitfield for EDDFR1 register, whether to be read from corresponding bitfield in AA64DFR1 register. Note: Only those bitfield which supports 'same as ID register or res0' functionality can be configured.

## **eddfr2\_reads\_idreg\_mask**

### Type

int

### Default value

0xffffffffffffffff

### Description

Mask to configure each bitfield for EDDFR2 register, whether to be read from corresponding bitfield in ID\_AA64DFR2 register. Note: Only those bitfield which supports 'same as ID register or res0' functionality can be configured.

## **eddfr\_reads\_idreg\_mask**

### Type

int

**Default value**

0xffffffffffffffff

**Description**

Mask to configure each bitfield for EDDFR register, whether to be read from corresponding bitfield in AA64DFR register. Note: Only those bitfield which supports 'same as ID register or res0' functionality can be configured.

**edpfr\_ras\_unknown\_bits\_read\_as\_0****Type**

bool

**Default value**

0x0

**Description**

If true then UNKNOWN bits in RAS field in EDPFR are read as 0.

**edxfr\_reads\_idreg****Type**

bool

**Default value**

0x0

**Description**

Whether EDDFR,EDDFR1 and EDDFR2 reads corresponding bitfield value from ID\_AA64DFR reg. Also, when this parameter is enabled, bitfields of these registers are configurable through 'eddf\*\_reads\_idreg\_mask' parameter.

**e10\_can\_access\_imp\_def\_functionality****Type**

bool

**Default value**

0x0

**Description**

If not made UNDEF by imp\_def\_functionality\_behaviour, ELO can access IMPLEMENTATION DEFINED registers and system instructions.

**e10\_e11\_only\_non\_secure****Type**

bool

**Default value**

0x0

**Description**

Secure/non-secure state if EL2 and EL3 are not implemented. 0, secure. 1, non-secure.

**e13\_trap\_priority\_when\_secure\_debug\_disabled****Type**

bool

**Default value**

0x0

**Description**

Undef when secure debug is disabled (EDSCR.SDD == 1) && boolean IMPLEMENTATION\_DEFINED 'EL3 trap priority when SDD == 1'.

**enable-gicv5****Type**

bool

**Default value**

0x0

**Description**

if enable-gicv5 is set, then GICv5 is Supported.

**enable\_address\_contig\_check****Type**

bool

**Default value**

0x0

**Description**

Check the input address range for the table entries that have the contiguous hint bit set.

**enable\_debug\_auth\_signals\_config****Type**

int

**Default value**

0xf

**Description**

Debug Authentication Signals DBGEN, SPIDEN (and if RME is enabled RLPIDEN and RTPIDEN) are configurable (default) or not configurable, (hardwired to 1). This parameter is the integer representation of a bitmap to enable configuration of these signals, with: - BIT[0] = DBGEN - BIT[1] = SPIDEN - BIT[2] = RLPIDEN - BIT[3] = RTPIDEN .



**enable\_tlb\_contig\_check****Type**

bool

**Default value**

0x0

**Description**

Perform extra pagetable walks to check translation table entries that have the contiguous hint bit set.

**enhanced\_pac2\_level****Type**

int

**Default value**

0x0

**Description**

Implements Enhanced PAC2 from ARMv8.6 (FEAT\_PAuth2), and PAC enhancements from ARMv9.5 (FEAT\_PAuth\_LR). options 0-3 of this feature are mandatory for ARMv8.6 but can be cherry-picked to a ARMv8.3(or greater) implementation. 0: No EnhancedPAC2, 1: EnhancedPAC2 Only (FEAT\_PAuth2), 2: EnhancedPAC2 with FPAC (FEAT\_FPAC), 3: EnhancedPAC2 with FPACCombined (FEAT\_FPACCOMBINE), 4: EnhancedPAC2 with LR signing (FEAT\_PAuth\_LR).

**error\_record\_feature\_register****Type**

string

**Default value**

""

**Description**

RAS feature register values. An array of JSON objects. The JSON schema for the array is: [{"ED":0x0, "IMPDEF\_3\_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other\_feature\_register\_values]. Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4. RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

**error\_record\_feature\_register\_json\_file****Type**

string

**Default value**

""

**Description**

File path to the RAS feature register values as JSON. The file uses the same format as the `error_record_feature_register` parameter value.

**`erxpfgctl_res0_stateful_mask`****Type**

int

**Default value**

0x0

**Description**

Mask for stateful bits for ERXPFGCTL which are RES0.

**`esr_write_update_res0`****Type**

bool

**Default value**

0x0

**Description**

If true, and RASv2 is enabled, then ESR\_ELx.WU field is RES0 for errors on both loads and stores (FEAT\_RASv2).

**`ete.ASYNC_PACKETS_WHEN_VIEWINST_OFF`****Type**

bool

**Default value**

0x0

**Description**

Generate the non-periodic alignment synchronisation packet generation when trace unit is operative.

**`ete.ATBTRIG`****Type**

bool

**Default value**

0x1

**Description**

ATB trigger support.

**ete.CCITMIN****Type**

int

**Default value**

0x4

**Description**

Minimum cycle count value.

**ete.CCSIZE****Type**

int

**Default value**

0xc

**Description**

Cycle counter size.

**ete.CLAIMTAGS****Type**

int

**Default value**

0x8

**Description**

Number of claim tags.

**ete.COMMOPT****Type**

bool

**Default value**

0x1

**Description**

Commit mode.

**ete.COMMTRANS****Type**

bool

**Default value**

0x0

**Description**

Commit transaction mode.

**ete.DEBUG****Type**

int

**Default value**

0x2

**Description**

DEBUG.

**ete.DESIGNER****Type**

int

**Default value**

0x41

**Description**

DESIGNER value.

**ete.ETE\_REVISION****Type**

int

**Default value**

0x0

**Description**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

**ete.EXCEPTION\_WITH\_CONTEXT****Type**

bool

**Default value**

0x1

**Description**

Whether EXCEPTION\_WITH\_CONTEXT packet is supported.

**ete.EXPLICITLY\_COMMIT\_PO\_ELEMS****Type**

bool

**Default value**

0x0

**Description**

Whether to unilaterally explicitly emit a commit after a PO packet.

**ete.IMPDEFEXCEPPERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of IMPDEF exceptions inserted in instruction blocks.

**ete.IMPDEF\_TRACE\_ON****Type**

int

**Default value**

0x0

**Description**

Whether trace is flushed and trace on packet generated by events described by bitmap value.  
bit 0 - PE entering low power state, bit 1 - PE entering debug state.

**ete.IMPRECISE\_FILTERING****Type**

int

**Default value**

0x0

**Description**

Number of instruction blocks traced on a transition in the filtering.

**ete.LPOVERRIDE****Type**

bool

**Default value**

0x1

**Description**

Low power override.

**ete.MAXSPEC****Type**

int

**Default value**

0x0

**Description**

Maximum speculation depth.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NOOVERFLOW****Type**

bool

**Default value**

0x0

**Description**

No overflow.

**ete.NUMACPAIRS****Type**

int

**Default value**

0x4

**Description**

Number of instruction address comparators pairs.

**ete.NUMCIDC****Type**

int

**Default value**

0x1

**Description**

Number of context ID comparators.

**ete.NUMCNTR****Type**

int

**Default value**

0x2

**Description**

Number of counters.

**ete.NUMEXTINSEL****Type**

int

**Default value**

0x4

**Description**

Number of external input selectors.

**ete.NUMPC****Type**

int

**Default value**

0x0

**Description**

Number of PE comparators.

**ete.NUMSEQSTATE****Type**

int

**Default value**

0x4

**Description**

Number of sequencer states.

**ete.NUMSSCC****Type**

int

**Default value**

0x1

**Description**

Number of single shot comparators.

**ete.NUMVMIDC****Type**

int

**Default value**

0x1

**Description**

Number of virtual ID comparators.

**ete.NumberOfETEEvents****Type**

int

**Default value**

0x2

**Description**

Number of trace events.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_DESIGNER****Type**

int

**Default value**

0x0

**Description**

TRCPIDR DESIGNER value.

**ete.PIDR\_PART****Type**

int

**Default value**

0x0



**Description**

TRCPIDR PART number value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.QFILT****Type**

bool

**Default value**

0x0

**Description**

Q filtering.

**ete.QSUP****Type**

int

**Default value**

0x0

**Description**

Q support.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RAZWI\_REG\_SEL\_TOP\_BIT**

**Type**

bool

**Default value**

0x0

**Description**

Implement Resource Selectors or Resource Selector Pairs bits as RAZ/WI.

**ete.REG\_ACCESS\_ONLY\_MODE**

**Type**

bool

**Default value**

0x0

**Description**

If enabled, all traces are disabled. Plugin only allows register acceses.

**ete.RES0\_STATEFUL**

**Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK**

**Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION**

**Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.STALLCTRL****Type**

bool

**Default value**

0x1

**Description**

Stall control.

**ete.SYSSTALL****Type**

bool

**Default value**

0x1

**Description**

System stall.

**ete.TRACEIDSIZE****Type**

int

**Default value**

0x7

**Description**

Trace ID size.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRACE\_OUTPUT\_ENABLE****Type**

bool

**Default value**

0x0

**Description**

ETE Trace output enable: 1=enable, 0=disable.

**ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.

**ete.TSMARK****Type**

bool

**Default value**

0x0

**Description**

Whether timestamp markers are supported.

**ete.TSSIZE****Type**

int

**Default value**

0x8

**Description**

Timestamp size.

**ete.WFXMODE****Type**

bool

**Default value**

0x1

**Description**

WFX mode.

**ets\_level****Type**

int

**Default value**

0x0

**Description**

Level of Enhanced Translation Synchronization support. 0: FEAT\_ETS2 not supported, 1: FEAT\_ETS2 not supported, 2: FEAT\_ETS2 supported, 3: FEAT\_ETS3 supported.

**exception\_catch\_before\_software\_step****Type**

bool

**Default value**

0x1

**Description**

Exception catch priority for the exception trapping form of exception catch (Armv8.2 or later, or exception\_catch\_type=0). If true, the exception catch debug event has higher priority than software step and halting step.

**exception\_catch\_type****Type**

int

**Default value**

0x0

**Description**

Type of exception catch (ARMv8.0 - ARMv8.1 only). 0, exception trapping. 1, non-exception trapping, higher priority than step. 2, non-exception trapping, lower priority than step.

**exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Exclusive monitors in the cluster will be cleared by a atomic by the same master to the monitored address.

**exclusive\_monitor\_clear\_on\_store\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Exclusive monitors in the cluster will be cleared by a store by the same master to the monitored address.

**exclusive\_monitor\_clear\_on\_strex\_address\_mismatch****Type**

bool

**Default value**

0x1

**Description**

Exclusive monitors in the cluster will be cleared when a strex fails because the address does not match.

**exclusive\_monitor\_clear\_on\_strex\_success****Type**

bool

**Default value**

0x1

**Description**

Exclusive monitors in the cluster will be cleared when a strex succeeds.

**exercise\_stxr\_fail****Type**

int

**Default value**

0x0

**Description**

Controls the rejection of exclusive store instructions. 0: exclusive store instructions should behave as normal, 1: Reject a pseudo-random majority of exclusive store instructions, 2: Always fail exclusive store instructions.

**ext\_abort\_device\_GRE\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_GRE\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_device\_GRE\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-GRE read external aborts.

**ext\_abort\_device\_GRE\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of device-GRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_read\_is\_sync.

**ext\_abort\_device\_GRE\_read\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_GRE\_read\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_device\_GRE\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-GRE write external aborts.

**ext\_abort\_device\_GRE\_write\_is\_sync****Type**

int

**Default value**

0x2



**Description**

Synchronous reporting of device-GRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_write\_is\_sync.

**ext\_abort\_device\_GRE\_write\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_GRE\_write\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_device\_nGRE\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_nGRE\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_device\_nGRE\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGRE read external aborts.

**ext\_abort\_device\_nGRE\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of device-nGRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_read\_is\_sync.

**ext\_abort\_device\_nGRE\_read\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_nGRE\_read\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_device\_nGRE\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGRE write external aborts.

**ext\_abort\_device\_nGRE\_write\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of device-nGRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_write\_is\_sync.

**ext\_abort\_device\_nGRE\_write\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_nGRE\_write\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_device\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_device\_read\_acquire\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device read with acquire external aborts.

**ext\_abort\_device\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_device\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGnRE write external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_device\_write\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_device\_write\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_fill\_data****Type**

int

**Default value**

0xfdfdfdfcfcfdfdfd

**Description**

Returned data, if external aborts are asynchronous.

**ext\_abort\_normal\_cacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal write-back cacheable-read external aborts.

**ext\_abort\_normal\_cacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of normal write-back cacheable-read external aborts.

**ext\_abort\_normal\_cacheable\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_cacheable\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_normal\_cacheable\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal write-back cacheable write external aborts.

**ext\_abort\_normal\_cacheable\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of normal write-back cacheable write external aborts.

**ext\_abort\_normal\_cacheable\_write\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_cacheable\_write\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_normal\_noncacheable\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_noncacheable\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_normal\_noncacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal noncacheable-read external aborts.

**ext\_abort\_normal\_noncacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1



**Description**

Synchronous reporting of normal noncacheable-read external aborts.

**ext\_abort\_normal\_noncacheable\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_noncacheable\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_normal\_noncacheable\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal noncacheable write external aborts.

**ext\_abort\_normal\_noncacheable\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of normal noncacheable write external aborts.

**ext\_abort\_normal\_noncacheable\_write\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_noncacheable\_write\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_normal\_wt\_cacheable\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_wt\_cacheable\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_normal\_wt\_cacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal write-through cacheable-read external aborts.

**ext\_abort\_normal\_wt\_cacheable\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of normal write-through read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_normal\_cacheable\_read\_is\_sync.

**ext\_abort\_normal\_wt\_cacheable\_read\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_wt\_cacheable\_read\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_normal\_wt\_cacheable\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of normal write-through write external aborts.

**ext\_abort\_normal\_wt\_cacheable\_write\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of normal write-through write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_normal\_cacheable\_write\_is\_sync.

**ext\_abort\_normal\_wt\_cacheable\_write\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_normal\_wt\_cacheable\_write\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_prefetch\_device\_GRE\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by device-GRE instruction fetches.

**ext\_abort\_prefetch\_device\_GRE\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by device-GRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_prefetch\_device\_nGRE\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by device-nGRE instruction fetches.

**ext\_abort\_prefetch\_device\_nGRE\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by device-nGRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_prefetch\_device\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by device-nGnRE instruction fetches.

**ext\_abort\_prefetch\_device\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by device-nGnRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_prefetch\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by normal writeback cacheable instruction fetches.

**ext\_abort\_prefetch\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Behaviour of external aborts generated by normal writeback cacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort.

**ext\_abort\_prefetch\_noncacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by normal noncacheable instruction fetches.

**ext\_abort\_prefetch\_noncacheable\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by normal noncacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_prefetch\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_prefetch\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_prefetch\_so\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by device-nGnRnE instruction fetches.

**ext\_abort\_prefetch\_so\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by device=nGnRnE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_prefetch\_wt\_cacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of external aborts generated by normal writethrough cacheable instruction fetches.

**ext\_abort\_prefetch\_wt\_cacheable\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Behaviour of external aborts generated by normal writethrough cacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

**ext\_abort\_so\_prefetch\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_so\_prefetch\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_so\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x1



**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_so\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_so\_write\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of device-nGnRnE write external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**ext\_abort\_so\_write\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_ttw\_cacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of TTW cacheable read external aborts.

**ext\_abort\_ttw\_cacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of TTW cacheable read external aborts.

**ext\_abort\_ttw\_cacheable\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_ttw\_cacheable\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**ext\_abort\_ttw\_noncacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of TTW noncacheable read external aborts.

**ext\_abort\_ttw\_noncacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of TTW noncacheable read external aborts.

**ext\_abort\_ttw\_noncacheable\_read\_ras\_index****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_ttw\_noncacheable\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_ttw\_wt\_cacheable\_read\_is\_critical****Type**

bool

**Default value**

0x0

**Description**

Critical reporting of TTW write-through cacheable read external aborts.

**ext\_abort\_ttw\_wt\_cacheable\_read\_is\_sync****Type**

int

**Default value**

0x2

**Description**

Synchronous reporting of TTW write-through cacheable read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_ttw\_cacheable\_read\_is\_sync.

**ext\_abort\_ttw\_wt\_cacheable\_read\_ras\_index****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_ttw\_cacheable\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

**ext\_abort\_ttw\_wt\_cacheable\_read\_ras\_type****Type**

int

**Default value**

0xffffffffffffffff

**Description**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_ttw\_cacheable\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**external\_debug\_request\_delay****Type**

int

**Default value**

0x0

**Description**

Configure External Debug Request delay in CPU cycles.

**external\_oslar\_access\_disabled\_by\_authentication****Type**

bool

**Default value**

0x0

**Description**

If true, external accesses to OSLAR, when external debugging is not enabled, will generate an error (FEAT\_Debugv8p2).

**fault\_on\_misprogrammed\_gpt\_contig\_region****Type**

bool

**Default value**

0x0

**Description**

Whether GPF faults occur when GPT contiguous entries are misprogrammed.

**fault\_on\_nT\_bit\_set****Type**

bool

**Default value**

0x1

**Description**

Whether block translation table entries with the nT bit set should always fault. Only applies when `changing_block_size_without_bbm_support_level` is 1 or higher.

**fault\_unalign\_to\_unsupported\_access****Type**

int

**Default value**

0x8

**Description**

If `has_unaligned_single_copy_atomics` is true, whether unaligned A64 atomic, exclusive and acquire/release instructions to non iWB-oWB or the access crossing a 16-byte boundary generate fault. Bits 0,1,2,3 should be set accordingly to enable the fault behaviour. bit 0: atomic access should fault, bit 1: exclusive access should fault, bit 2: acquire/release should fault, bit 3: the access crossing a 16-byte boundary should fault.

**`fault_unaligned_s1_device_s2_fwb`****Type**

int

**Default value**

0x0

**Description**

Whether unaligned fault with stage1 Device memory and final memory attribute forced to normal by FWB. 0, No fault. 1, will fault. 2 No fault if final Shareability is NSH.

**`force_align_pc`****Type**

bool

**Default value**

0x0

**Description**

UNPREDICTABLE branch to non-word-aligned address in ARM state is forced to be aligned.

**`force_deterministic_irg_tag_generation`****Type**

bool

**Default value**

0x0

**Description**

Force the random tag generated by the IRG instruction when `GCR_EL1.RRND=1` to equal `RGSR_EL1.SEED[3:0]` rather than a non-deterministic value.

**`force_mte_tag_access_razwi_and_ignore_tag_checks`****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation

instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_pstate\_pm\_in\_debug\_state****Type**

bool

**Default value**

0x0

**Description**

If true, PSTATE.PM is forced to 1 while entering in debug state (FEAT\_EBEP).

**force\_sync\_on\_wfx****Type**

bool

**Default value**

0x0

**Description**

If true, the PE does a context synchronization before entering low power state(WFI/WFE).

**force\_wnr\_read\_unsupported\_exclusive\_or\_atomic****Type**

bool

**Default value**

0x0

**Description**

DEPRECATED, please use force\_wnr\_read\_unsupported\_exclusive\_or\_atomic instead.  
Whether ESR\_ELx.WnR is forced to 0 for unsupported atomic and exclusives.

**force\_wnr\_unsupported\_atomic\_hwu****Type**

int

**Default value**

0x0

**Description**

If not 0, force ESR\_ELx.WnR to a specific value on Unsupported Atomic Hardware Update.  
Possible values: 0: Not forced. 1: Write. 2: Read.

**force\_wnr\_unsupported\_exclusive\_or\_atomic****Type**

int

**Default value**

0x0

**Description**

If not 0, force ESR\_ELx.WnR to a specific value on Unsupported Atomic or Exclusives.  
Possible values: 0: Not forced. 1: Write. 2: Read.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect.  
1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**fp8\_support\_level****Type**

int

**Default value**

0x0

**Description**

0->No support for Advanced SIMD, SVE2 FP8 instructions 1->Support for FEAT\_FP8  
2->Support for FEAT\_FP8FMA 3->Support for FEAT\_FP8DOT4 4->Support for  
FEAT\_FP8DOT2.

**fpcr\_short\_vector\_raz****Type**

bool

**Default value**

0x0



**Description**

FPSCR and FPCR fields LEN and STRIDE are hardwired to 0.

**fpsr\_res0\_stateful\_mask****Type**

int

**Default value**

0x0

**Description**

Mask for stateful bits of FPSR which are RES0.

**fsr\_ext\_bit\_update\_kind****Type**

int

**Default value**

0x3

**Description**

Set/Clear DFSR/IFSR EA bit on Synchronous/Async External Aborts. 0: Never Set, 1: Set on Synchronous Ext Aborts 2: Set on Asynchronous Ext Aborts 3: Set on both Sync and Async Ext Aborts.

**gcs\_data\_check\_overrides\_data\_abort****Type**

bool

**Default value**

0x0

**Description**

GCS Data check exceptions are taken before Data Aborts.

**gcs\_overshoot\_writes****Type**

int

**Default value**

0x0

**Description**

Number of overshooting GCS records written after a writing a record.

**gcspr\_sync\_immediate****Type**

bool

**Default value**

0x0

**Description**

If true, writing to GCSPR\_ELx registers has immediate effect regardless of has\_delayed\_sysreg flag.

**gic.GICC-offset****Type**

int

**Default value**

0x2000

**Description**

Offset from PERIPHBASE for GICC registers.

**gic.GICD-offset****Type**

int

**Default value**

0x1000

**Description**

Offset from PERIPHBASE for GICD registers. Will be ignored when GICv3 CPU interface is enabled, as distributor is then external to the cluster.

**gic.GICH-offset****Type**

int

**Default value**

0x4000

**Description**

Offset from PERIPHBASE for GICH registers.

**gic.GICH-other-CPU-offset****Type**

int

**Default value**

0x5000

**Description**

Offset from PERIPHBASE for GICH registers for accessing other CPUs in the cluster. Set to 0 to disable.

**gic.GICV-alias****Type**

int

**Default value**

0x0

**Description**

Offset from PERIPHBASE for alias of GICV registers. When gicv2-only, if zero no alias will be created; if gicv2-only=0, the param is deprecated, when zero or unset an alias is created in the place mandated by the architecture (GICV-base+0xF000).

**gic.GICV-offset****Type**

int

**Default value**

0x6000

**Description**

Offset from PERIPHBASE for GICV registers.

**gic.PERIPH-size****Type**

int

**Default value**

0x8000

**Description**

Size of registers based at PERIPHBASE that are considered to be owned by the GIC. Any accesses in the range PERIPHBASE to PERIPHBASE+gic.PERIPH-size-1 that do not match GIC registers will be treated as RAZ/WI.

**gicv3.A3-affinity-supported****Type**

bool

**Default value**

0x0

**Description**

Whether a non-zero value for affinity at level 3 is supported.

**gicv3.BPR-min****Type**

int

**Default value**

0x2

**Description**

The minimum value for the GICC\_BPR register (non-secure version will be 1 + this value).

**`gicv3.EOI-check-CPUID`****Type**

bool

**Default value**

0x0

**Description**

Check CPU ID specified for accesses to EOI registers (rather than just ending highest priority active interrupt).

**`gicv3.EOI-check-ID`****Type**

bool

**Default value**

0x0

**Description**

Check Interrupt ID specified for accesses to EOI registers (rather than just ending highest priority active interrupt).

**`gicv3.EOI-deactivate-any-interrupt`****Type**

bool

**Default value**

0x0

**Description**

Allow an EOI to deactivate interrupts that aren't the highest priority active interrupt (EOI-ignore-out-of-order must be false otherwise this is ignored).

**`gicv3.EOI-ignore-out-of-order`****Type**

bool

**Default value**

0x1

**Description**

Ignore EOI writes that cannot end the highest priority active interrupt.

**gicv3.FIQEn-RAO****Type**

bool

**Default value**

0x0

**Description**

GICC\_CTLR.FIQEn is read as one, write insensitive.

**gicv3.IIDR\_base****Type**

int

**Default value**

0x43b

**Description**

The base value for calculating the GICC\_IIDR register value.

**gicv3.LR-count****Type**

int

**Default value**

0x10

**Description**

The number of implemented list registers.

**gicv3.PMHE-RAO-WI****Type**

bool

**Default value**

0x0

**Description**

ICC\_CTLR\_EL\*.PHME is read as one, write insensitive.

**gicv3.PMHE-RAZ-WI****Type**

bool

**Default value**

0x0

**Description**

ICC\_CTLR\_EL\*.PHME is read as zero, write insensitive.

**gicv3.PMHE-release-set-packet****Type**

bool

**Default value**

0x0

**Description**

if PHME is enabled, whether a SET packet is released by CPU Intf in Upstream Ack window.

**gicv3.SRE-EL2-enable-RAO****Type**

bool

**Default value**

0x0

**Description**

When ICC\_SRE\_EL2.SRE is RAO/WI, makes ICC\_SRE\_EL2.Enable RAO/WI.

**gicv3.SRE-EL3-enable-RAO****Type**

bool

**Default value**

0x0

**Description**

When ICC\_SRE\_EL3.SRE is RAO/WI, makes ICC\_SRE\_EL3.Enable RAO/WI.

**gicv3.SRE-EL3-set-once****Type**

bool

**Default value**

0x0

**Description**

Restrict SRE EL3 to be set only once.

**gicv3.SRE-enable-action-on-mmap****Type**

int

**Default value**

0x0

**Description**

Allowed values are: 0-SRE one allows mmap access. 1-SRE one disables mmap access. 2-SRE one makes mmap access RAZ-WI.

**`gicv3.STATUSR-implemented`****Type**

bool

**Default value**

0x1

**Description**

If GICv3 CPU interface is being used, this determines whether the STATUS registers are implemented.

**`gicv3.VBPR-min`****Type**

int

**Default value**

0x2

**Description**

The minimum value for the GICV\_BPR register (non-secure version will be 1 + this value).

**`gicv3.VFIQEn-RAO`****Type**

bool

**Default value**

0x0

**Description**

ICH\_VMCR\_EL2.VFIQEn is read as one, write insensitive.

**`gicv3.cuintf-mmap-access-level`****Type**

int

**Default value**

0x0

**Description**

Allowed values are: 0-mmap access is supported for GICC,GICH,GICV registers. 1-mmap access is supported only for GICV registers. 2-mmap access is not supported.

**gicv3.dir-trap-support****Type**

bool

**Default value**

0x1

**Description**

The cpu supports separate trapping of ICC\_DIR\_EL1 to EL2.

**gicv3.el3\_trap\_priority\_when\_secure\_debug\_disabled****Type**

bool

**Default value**

0x0

**Description**

Undef to access priorities group register when secure debug is disabled.

**gicv3.extended-interrupt-range-support****Type**

bool

**Default value**

0x0

**Description**

Device has support for extended SPI/PPI ID ranges.

**gicv3.gicv2-only****Type**

bool

**Default value**

0x0

**Description**

Limit the GIC implementation to GICv2 features only.

**gicv3.idle-is-ff****Type**

bool

**Default value**

0x1

**Description**

For GICC/GICV RPR, when idle, return FF when true, minimum supported priority otherwise.



**gicv3.ignore-DIR-write-when-EOImode-not-set****Type**

bool

**Default value**

0x1

**Description**

Ignore UNPREDICTABLE access to GICC\_DIR register.

**gicv3.interrupt-bypass-support****Type**

bool

**Default value**

0x1

**Description**

Interrupt bypass support, set to false for devices not supporting interrupt bypass.

**gicv3.local-SEIs****Type**

bool

**Default value**

0x0

**Description**

Generate SEI to signal internal issues.

**gicv3.local-VSEIs****Type**

bool

**Default value**

0x0

**Description**

Generate VSEI to signal internal issues.

**gicv3.physical-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of physical ID bits implemented.

**gicv3.priority-bits****Type**

int

**Default value**

0x5

**Description**

Number of priority bits implemented.

**gicv3.send-PMHE-command-only-when-priority-changes****Type**

bool

**Default value**

0x0

**Description**

Send PMHE upstream command to distributor only when write to ICC\_PMR\_EL1 changes the priority.

**gicv3.sgi-range-selector-support****Type**

bool

**Default value**

0x0

**Description**

Device has support for the Range Selector feature for SGI.

**gicv3.suppress-virtual-enables-comms****Type**

bool

**Default value**

0x1

**Description**

In GICv3 only mode, prevents the GIC CPUIF from communicating UpstreamWrite/VirtualEnables to the IRI.

**gicv3.virtual-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of virtual ID bits implemented.

**`gicv3.virtual-lpi-support`****Type**

bool

**Default value**

0x1

**Description**

When GICv3 is supported, indicates a cut down CPUIF interface with no support of VLPI (GICv3 only) when false.

**`gicv3.virtual-priority-bits`****Type**

int

**Default value**

0x5

**Description**

Number of virtual priority bits implemented.

**`gicv3.without-DS-support`****Type**

bool

**Default value**

0x0

**Description**

GICv3 CPU interfaces do not support disabling security in the distributor (GICD\_CTLR.DS=1).

**`gicv4.mask-virtual-interrupt`****Type**

bool

**Default value**

0x0

**Description**

If true, virtual interrupts can be masked from being reported to virtual CPU interface by setting ICH\_HCR\_EL2.DVIM 1. No control otherwise.

**gicv5.config\_file****Type**

string

**Default value**

""

**Description**

File path for the GICv5 configuration yaml. The file lists the GICv5 params.

**gicv5.interrupt-bypass-support****Type**

bool

**Default value**

0x0

**Description**

Interrupt bypass support. when set to true, bypasses GICv5 CPU interface to signal interrupts to the PE.

**global\_debug\_rom.ROMDEVID****Type**

int

**Default value**

0x0

**Description**

Value of Debug Rom Device Identification Register.

**global\_debug\_rom.ROMPIDR****Type**

int

**Default value**

0x4000bb000

**Description**

Value of Debug Rom Peripheral Identification Register.

**global\_debug\_rom.ROMPRIDR0****Type**

int

**Default value**

0x1

**Description**

Value of Debug ROM Power RequestID Register.

**gpccr\_el3\_gpcp\_behaviour****Type**

int

**Default value**

0x2

**Description**

Used to control impdef behaviour when GPCP=1 (0->Faults are always generated and reported, 1->Faults are not generated and reported), 2->Faults are generated and reported only for Arm recommended cases.

**gpt\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of separate GPT TLB entries.

**gpt\_walkcache\_size****Type**

int

**Default value**

0x0

**Description**

Number of GPT walk cache entries.

**hardware\_translation\_table\_update\_implemented****Type**

int

**Default value**

0x1

**Description**

Implement hardware translation table updates from ARMv8.1 (FEAT\_HAFDBS). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.1 is enabled. - 2, feature is implemented.

**has-gicv4.1****Type**

bool

**Default value**

0x0

**Description**

GICv4.1 is enabled, and all the features with GICv4.1 are implemented (FEAT\_GICv4p1).

**has\_128\_bit\_atomic\_instructions****Type**

int

**Default value**

0x0

**Description**

Implement 128-bit Atomic Instructions (FEAT\_LSE128) Possible values of this parameter are:  
- 0, feature is not enabled.  
- 1, feature is implemented if ARMv9.4 is enabled.  
- 2, feature is implemented.

**has\_128\_bit\_tt\_descriptors****Type**

int

**Default value**

0x0

**Description**

Implement 128-bit Translation Table Descriptors (FEAT\_D128) Possible values of this parameter are:  
- 0, feature is not enabled.  
- 1, feature is implemented if ARMv9.4 is enabled.  
- 2, feature is implemented.

**has\_16bit\_asids****Type**

bool

**Default value**

0x1

**Description**

Enable 16-bit ASIDs.

**has\_16bit\_vmids****Type**

int

**Default value**

0x1

**Description**

Implement support for 16-bit VMIDs from ARMv8.1 (FEAT\_VMID16). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.1 is enabled. - 2, feature is implemented.

**has\_16k\_granule****Type**

bool

**Default value**

0x0

**Description**

Implement the 16k LPAE translation granule.

**has\_4k\_granule****Type**

bool

**Default value**

0x1

**Description**

Implement the 4k LPAE translation granule.

**has\_52bit\_address\_with\_16k****Type**

int

**Default value**

0x0

**Description**

Implements Armv8.7 52-bit IPA/PA support for 16k (FEAT\_LPA2). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_52bit\_address\_with\_4k****Type**

int

**Default value**

0x0

**Description**

Implements Armv8.7 52-bit IPA/PA support for 4k (FEAT\_LPA2). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_56\_bit\_va****Type**

int

**Default value**

0x0

**Description**

56-bit Physical Address, identified as (FEAT\_LVA3) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.4 is enabled. - 2, feature is implemented.

**has\_64bit\_pmu\_ext\_access****Type**

int

**Default value**

0x0

**Description**

Implement 64-bit pmu external interface access (FEAT\_PMU\_EXT64) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_64k\_granule****Type**

bool

**Default value**

0x1

**Description**

Implement the 64k LPAE translation granule.

**has\_aarch32\_dbgdidr\_etc****Type**

bool

**Default value**

0x1

**Description**

DBGDIDR, DBGDRAR, DBGDSAR exist even if EL1 doesn't implement AArch32.



**has\_aarch32\_hpd****Type**

bool

**Default value**

0x0

**Description**

If true then hierarchical permission disable is supported in AArch32 (FEAT\_AA32HPD).

**has\_aarch64****Type**

bool

**Default value**

0x1

**Description**

All implemented exception levels can run in AArch64.

**has\_actlr2****Type**

bool

**Default value**

0x0

**Description**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

**has\_actlr\_virtualisation****Type**

bool

**Default value**

0x0

**Description**

If true ACTLR\_EL12 is implemented and ACTLR\_EL1 supports virtualisation.

**has\_address\_breakpoint\_linking****Type**

int

**Default value**

0x0

**Description**

Implement Address Breakpoint Linking Extension (FEAT\_ABLE) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.4 is enabled. - 2, feature is implemented.

**has\_amu****Type**

int

**Default value**

0x0

**Description**

Implement activity monitor functionality from ARMv8.4 (FEAT\_AMUv1). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_amu\_ext64****Type**

int

**Default value**

0x0

**Description**

Implement 64-bit external interface to the Activity Monitors (FEAT\_AMU\_EXT64). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_amu\_extacr****Type**

bool

**Default value**

0x0

**Description**

If true then implement AMU external access control registers (FEAT\_AMU\_EXTACR).

**has\_arm\_v8-1****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8.1 Extension.

**has\_arm\_v8-2****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8.2 Extension.

**has\_arm\_v8-3****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8.3 Extension.

**has\_arm\_v8-4****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8.4 Extension.

**has\_arm\_v8-5****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8.5 Extension.

**has\_arm\_v8-6****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8.6 Extension.

**has\_arm\_v8-7****Type**

bool

**Default value**

0x0

**Description**

Implement the Armv8.7 Extension.

**has\_arm\_v8-8****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8.8 Extension.

**has\_arm\_v8-9****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8.9 Extension.

**has\_arm\_v9-0****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv9.0 Extension.

**has\_arm\_v9-1****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv9.1 Extension.

**has\_arm\_v9-2****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv9.2 Extension.

**has\_arm\_v9-3****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv9.3 Extension.

**has\_arm\_v9-4****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv9.4 Extension.

**has\_arm\_v9-5****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv9.5 Extension.

**has\_arm\_v9-6****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv9.6 Extension.

**has\_asid2****Type**

int

**Default value**

0x1

**Description**

If true then support for use of two concurrent ASIDs (FEAT\_ASID2). Possible values of this parameter are: - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_at\_with\_pan****Type**

int

**Default value**

0x1

**Description**

Implement new AT instructions with PAN support (FEAT\_PAN2). Possible values of this parameter are: - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_ats1a****Type**

int

**Default value**

0x0

**Description**

Support for ATS1ExR instructions (FEAT\_ATS1A). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_attribute\_index\_enhancement****Type**

int

**Default value**

0x0

**Description**

Memory Attribute Index Enhancement (FEAT\_AIE). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_axflag\_xaflag****Type**

int

**Default value**

0x1

**Description**

Implement flag manipulation instructions (AXFlag, XAFlag) from ARMv8.5 (FEAT\_FlagM2). Possible values of this parameter are: - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_axflag\_xaflag\_frint****Type**

int

**Default value**

0x1

**Description**

Implement flag manipulation instructions (AXFlag, XAFlag) and floating-point rounding to int instructions (FRINT[32|64][X|Z]) from ARMv8.5. If this parameter is enabled, it also enables both has\_axflag\_xaflag and has\_frint. If support for only one of the features is needed, please use the individual parameters and do not enable this one (FEAT\_FlagM2, FEAT\_FRINTTS). Possible values of this parameter are: - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_bc****Type**

int

**Default value**

0x1

**Description**

Implement Armv8.8 Hinted Conditional Branch (FEAT\_HBC) Possible values of this parameter are: - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_branch\_target\_exception****Type**

int

**Default value**

0x1

**Description**

Implement Branch target identification mechanism from ARMv8.5 (FEAT\_BTI). Possible values of this parameter are: - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_brbe****Type**

bool

**Default value**

0x0

**Description**

If true, implements branch record buffer extension (FEAT\_BRBE).

**has\_brbe\_v1p1****Type**

bool

**Default value**

0x0

**Description**

If true, implements FEAT\_BRBEv1p1.

**has\_ccidx****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8.3 CCSIDR Extension. Extending the ccsidr number of sets.

**has\_cfinv\_rmif\_setf****Type**

int

**Default value**

0x1

**Description**

Implement flag manipulation (CFINV, RMIF, SETF8, SETF16) instructions from ARMv8.4 (FEAT\_FlagM). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_chkfeat****Type**

int

**Default value**

0x0



**Description**

Implement CHKFEAT instruction from ARMv9.4 (FEAT\_CHK). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.4 is enabled. - 2, feature is implemented.

**has\_clear\_bhb****Type**

int

**Default value**

0x1

**Description**

Implement Clear Branch History information instruction (FEAT\_CLRBHB). Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_clear\_other\_speculation\_by\_context****Type**

int

**Default value**

0x1

**Description**

Implement execution and data prediction invalidation from Armv8.9 (FEAT\_SPECRES2). Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_cmo\_wr\_control****Type**

int

**Default value**

0x1

**Description**

Whether stage1/2 CMO write perm control is supported (FEAT\_CMOW) Possible values of this parameter are: - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_cmpbr****Type**

int

**Default value**

0x0

**Description**

Implement compare and branch instructions from ARMv9.6, optional in ARMv9.5 (FEAT\_CMPBR). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_coherent\_icache****Type**

bool

**Default value**

0x0

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_common\_not\_private\_translations****Type**

int

**Default value**

0x1

**Description**

Implement the TTBRn\_ELx.CnP (Common not Private) controls from ARMv8.2 (FEAT\_TTCNP). Possible values of this parameter are: - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_complex\_number****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.3 complex number support, Multiply Accumulate and Add instructions (FEAT\_FCMA). Possible values of this parameter are: - 1, feature is implemented if ARMv8.3 is enabled. - 2, feature is implemented.

**has\_const\_pac****Type**

int

**Default value**

0x0

**Description**

Feature for singular selection of PAC field (FEAT\_CONSTPACFIELD). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_cssc****Type**

int

**Default value**

0x1

**Description**

Support for common short sequence compression instructions (FEAT\_CSSC) Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_cvadp\_support****Type**

int

**Default value**

0x0

**Description**

Implement instruction to support cache clean by deep persistence (DC CVADP) from ARMv8.5, can be selected for core implemented on any arch version starting ARMv8.2 (FEAT\_DPB, FEAT\_DPB2). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_data\_alignment\_flag****Type**

int

**Default value**

0x0

**Description**

Implement non-optimal misalignment flag for PMU/SPE from ARMv8.5 Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_debug\_rom****Type**

bool

**Default value**

0x1

**Description**

If true, a debug ROM will be generated describing the cluster's debug components.

**has\_delayed\_brbe\_records****Type**

bool

**Default value**

0x1

**Description**

If true, a synchronization barrier is required to update the BRB records (FEAT\_BRBE).

**has\_delayed\_ctireg****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of CTI register writes until ISB or implicit barrier.

**has\_delayed\_dbgreg****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

**has\_delayed\_mdscr\_el1****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of MDSCR\_EL1 register writes until ISB or implicit barrier.

**has\_delayed\_oslar\_el1****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of OSLAR\_EL1 register writes until ISB or implicit barrier.

**has\_delayed\_pmureg****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of PMU register writes until ISB or implicit barrier.

**has\_delayed\_sysreg****Type**

bool

**Default value**

0x0

**Description**

Delay the functional effect of system register writes until ISB or implicit barrier.

**has\_delayed\_wfe\_trap****Type**

int

**Default value**

0x0

**Description**

Implements Configurable Delayed WFE trapping from ARMv8.6 (FEAT\_TWED). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**has\_dgh****Type**

int

**Default value**

0x1

**Description**

Implements Data Gathering Hint instruction from ARMv8.6 (FEAT\_DGH). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**has\_dot\_product****Type**

int

**Default value**

0x1

**Description**

Implement the dot product (UDOT, SDOT) instructions from ARMv8.4 (FEAT\_DotProd). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_e0pd****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.5 feature to prevent unprivileged access to one half of the memory (FEAT\_E0PD). Possible values of this parameter are: - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_e2h\_rao****Type**

int

**Default value**

0x0

**Description**

Whether the implementation treats HCR\_EL2.E2H as Read-As-One (RAO). 0 : FEAT\_E2H0 implemented Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.1 is enabled. - 2, feature is implemented.

**has\_ebep****Type**

int

**Default value**

0x0

**Description**

Implement Exception-Based Event Profiling from ARMv9.4 (FEAT\_EBEP). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.4 is enabled. - 2, feature is implemented.

**has\_ebf16****Type**

int

**Default value**

0x0

**Description**

Support for Extended BFloat16 Behaviours (FEAT\_EBF16) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_ecbhb****Type**

int

**Default value**

0x1

**Description**

Implement Exploitative Control using Branch History information between exception levels (FEAT\_ECBHB). Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_edacr****Type**

bool

**Default value**

0x1

**Description**

Implement EDACR register.

**has\_edhsr****Type**

int

**Default value**

0xf

**Description**

Implement external debug halt status register (FEAT\_EDHSR). 0: FEAT\_EDHSR is not implemented unless architecturally required by another feature, 1: FEAT\_EDHSR is implemented, 2: FEAT\_EDHSR is implemented (extends EDHSR to include the VNCR, CM, and WnR fields), 0xF: FEAT\_EDHSR implementation is dependent on FEAT\_SME.

**has\_el2****Type**

bool

**Default value**

0x1

**Description**

Implements EL2.

**has\_el3****Type**

bool

**Default value**

0x1

**Description**

Implements EL3.

**has\_enhanced\_pac****Type**

bool

**Default value**

0x0

**Description**

If pointer authentication is enabled then implement enhanced PAC (FEAT\_EPAC).

**has\_enhanced\_pan****Type**

int

**Default value**

0x1

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_enhanced\_software\_step****Type**

int

**Default value**

0x1



**Description**

Implement Enhanced Software Step Extension (FEAT\_STEP2). Possible values of this parameter are: - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_ets****Type**

bool

**Default value**

0x0

**Description**

Whether Enhanced Translation Synchronization is supported. (NOTE: DEPRECATED: use ets\_level instead).

**has\_exception\_trapping\_form\_of\_vector\_catch****Type**

bool

**Default value**

0x1

**Description**

Implement the exception trapping form of vector catch debug event.

**has\_extended\_recip\_estimate****Type**

int

**Default value**

0x0

**Description**

Implements increased precision of reciprocal instructions (FEAT\_RPRES). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_external\_rndr****Type**

int

**Default value**

0x0

**Description**

Implement external random number generator module. When enabling this with has\_rndr enabled, the external random number generator will be used instead of internal random number generator. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_f8f16mm****Type**

bool

**Default value**

0x0

**Description**

Implement FEAT\_F8F16MM and dependent features.

**has\_f8f32mm****Type**

bool

**Default value**

0x0

**Description**

Implement FEAT\_F8F32MM and dependent features.

**has\_faminmax****Type**

bool

**Default value**

0x0

**Description**

Implement FEAT\_FAMINMAX.

**has\_far\_not\_valid****Type**

bool

**Default value**

0x0

**Description**

Implements FnV bit in ESR\_ELx and xFSR, FAR not valid for synchronous external aborts.

**has\_far\_not\_valid\_dfsc****Type**

bool

**Default value**

0x0

**Description**

Implements FnV bit in ESR\_ELx, FAR not valid for synchronous external aborts for Data Abort.

**has\_far\_not\_valid\_ifsc****Type**

bool

**Default value**

0x0

**Description**

Implements FnV bit in ESR\_ELx and xFSR, FAR not valid for synchronous external aborts for Instruction Abort.

**has\_feat\_pops****Type**

bool

**Default value**

0x0

**Description**

Whether ARMv9.6 RAS support for clean-and-invalidate of data by virtual address to Point of Physical Storage (FEAT\_PoPS).

**has\_fgt****Type**

int

**Default value**

0x1

**Description**

Implements Fine-grained Virtualization Traps extension from ARMv8.6 (FEAT\_FGT). Possible values of this parameter are: - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**has\_fgt2****Type**

int

**Default value**

0x1

**Description**

Implement additional FGT traps introduced in ARMv8.9 (FEAT\_FGT2). Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_fgwt3****Type**

bool

**Default value**

0x0

**Description**

If true then Fine Grained Write EL3 is enabled (FEAT\_FGWTE3).

**has\_fixed\_function\_instr\_counter****Type**

int

**Default value**

0x0

**Description**

Implement fixed-function instruction counter (FEAT\_PMUv3\_ICNTR) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_fp16****Type**

int

**Default value**

0x1

**Description**

Implement the half-precision floating-point data processing instructions from ARMv8.2 (FEAT\_FP16). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_fp16\_fmlal****Type**

int

**Default value**

0x1

**Description**

Implement the New Floating Point Multiplication Variant (FP16 FMLAL, FMLSL) instructions from ARMv8.4. Only supported if has\_fp16=0x1 (FEAT\_FHM). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_fpmr****Type**

bool

**Default value**

0x0

**Description**

Implement FPMR (FEAT\_FPMR).

**has\_fprcvt****Type**

int

**Default value**

0x1

**Description**

Implement FEAT\_FPRCVT FP convert instructions from ARMv9.6, optional in ARMv9.5 (FEAT\_FPRCVT). Possible values of this parameter are: - 1, feature is implemented if ARMv9.6 is enabled. - 2, feature is implemented.

**has\_frint****Type**

int

**Default value**

0x1

**Description**

Implement floating-point rounding to int instructions (FRINT[32|64][X|Z]) from ARMv8.5 (FEAT\_FRINTTS). Possible values of this parameter are: - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_gcs****Type**

int

**Default value**

0x0

**Description**

Implement Guarded Control Stack Extension from ARMv9.4 (FEAT\_GCS). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.4 is enabled. - 2, feature is implemented.

**has\_generic\_authentication****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.3 generic authentication. Possible values of this parameter are: - 1, feature is implemented if ARMv8.3 is enabled. - 2, feature is implemented.

**has\_guest\_translation\_granule****Type**

int

**Default value**

0x1

**Description**

Implement mechanism for guest translation granule identification from ARMv8.5, ID values determined by stage1 granule configuration parameters (FEAT\_GTG). Possible values of this parameter are: - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_haft****Type**

int

**Default value**

0x0

**Description**

Implement Hardware managed Access Flag for Table Descriptors (FEAT\_HAFT) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_hardware\_accelerator\_for\_cleaning\_dirty\_state****Type**

int

**Default value**

0x0

**Description**

Whether hardware accelerator for cleaning Dirty state is supported (FEAT\_HACDBS).

Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_hardware\_dirty\_state\_tracking\_structure****Type**

int

**Default value**

0x0

**Description**

Whether hardware Dirty state tracking Structure is supported (FEAT\_HDBSS). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_hardware\_translation\_table\_update****Type**

int

**Default value**

0x2

**Description**

Type of hardware translation table supported (when enabled by hardware\_translation\_table\_update\_implemented). 0, not implemented. 1, access bit updates implemented. 2, access bit updates and dirty bit mechanism implemented (FEAT\_HAFDBS).

**has\_hcrx\_el2****Type**

int

**Default value**

0x1

**Description**

Implements new HCRX\_EL2 id register from Armv8.7 (FEAT\_HCX). Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_hpmn0****Type**

int

**Default value**

0x0

**Description**

Allow hypervisor to set MDCR\_EL2.HPMN to 0 (FEAT\_HPMN0) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_id\_reg\_read****Type**

int

**Default value**

0x1

**Description**

Implement read access to the ID registers (ESR\_ELx.EC=0x18) (FEAT\_IDST) Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_iesb****Type**

int

**Default value**

0x1

**Description**

Implement support for implicit error sync event from ARMv8.2 (FEAT\_IESB). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_itd****Type**

bool

**Default value**

0x1

**Description**

Implement the optional IT disable feature.

**has\_ite****Type**

int

**Default value**

0x0



**Description**

Implement Instrumentation Trace Extension from ARMv9.4 (FEAT\_ITE). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.4 is enabled. - 2, feature is implemented.

**has\_jscvt****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.3 javascript Floating-point to Integer conversion instruction (FEAT\_JSCVT). Possible values of this parameter are: - 1, feature is implemented if ARMv8.3 is enabled. - 2, feature is implemented.

**has\_large\_system\_ext****Type**

bool

**Default value**

0x0

**Description**

Implement the ARMv8 Large System Extensions (FEAT\_LSE).

**has\_large\_ttbr\_ba\_without\_lpa****Type**

bool

**Default value**

0x1

**Description**

When FEAT\_LPA is not implemented, whether TTBR base address supports large values (52 bits) or not (48 bits).

**has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_ldapur\_stlur****Type**

int

**Default value**

0x1

**Description**

Implement support for LDAPR and STLUR instructions with immediate offsets from ARMv8.4 (FEAT\_LRCPC2). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_ldm\_stm\_ordering\_control****Type**

int

**Default value**

0x0

**Description**

Implement the SCTLRL\_ELx.LSMAOE (Load/Store Multiple Atomicity and Ordering Enable) and SCTLRL\_ELx.nTLSMD (no Trap Load/Store Multiple to Device) controls from ARMv8.2 (FEAT\_LSMAOC). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_lrcpc****Type**

bool

**Default value**

0x0

**Description**

If true then it support the RCpc feature from ARMv8.3 (FEAT\_LRCPC).

**has\_lrcpc3****Type**

int

**Default value**

0x0

**Description**

Implement Release Consistency processor consistent (RCpc) feature from Armv8.9 (FEAT\_LRCPC3). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_lsfe****Type**

int

**Default value**

0x0

**Description**

Implement A64 base Atomic floating-point in-memory instructions (FEAT\_LSFE) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.3 is enabled. - 2, feature is implemented.

**has\_lsui****Type**

int

**Default value**

0x1

**Description**

Implement additional load and store unprivileged instructions (FEAT\_LSUI). Possible values of this parameter are: - 1, feature is implemented if ARMv9.6 is enabled. - 2, feature is implemented.

**has\_lut****Type**

bool

**Default value**

0x0

**Description**

Implement FEAT\_LUT.

**has\_mbist\_never1\_ae25****Type**

bool

**Default value**

0x0

**Description**

Implements the MBIST Never1 detection for AE25 class cpu.

**has\_mismatch\_and\_range\_breakpoints****Type**

int

**Default value**

0x0

**Description**

Implement Mismatch and Range Breakpoints (FEAT\_BWE) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.4 is enabled. - 2, feature is implemented.

**has\_mismatch\_watchpoints****Type**

int

**Default value**

0x0

**Description**

Implement Breakpoints and Watchpoints Enhancements (FEAT\_BWE2). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_mops\_option****Type**

int

**Default value**

0x0

**Description**

Implement Armv8.8 standard instructions for memory operations (FEAT\_MOPS). 0, not implemented (unsupported if Armv8.8 is enabled). 1, implemented using Option A. 2, implemented using Option B.

**has\_mpam****Type**

int

**Default value**

0x0

**Description**

Implement ARMv8.4 MPAM Registers and associated functionality (FEAT\_MPAM). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_mpm****Type**

bool

**Default value**

0x0

**Description**

Implement max-power mitigation mechanism (MPMM).

**has\_mt\_pmu\_disable\_feature****Type**

int

**Default value**

0x0

**Description**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

**has\_mte\_async\_faults****Type**

bool

**Default value**

0x1

**Description**

Whether MTE asynchronous faults are supported (FEAT\_MTE\_ASYNC).

**has\_mte\_ctrl\_bits\_stateful****Type**

bool

**Default value**

0x0

**Description**

if memory\_tagging\_support\_level == 1, Whether mte specific control bits in system registers are stateful.

**has\_mte\_perm****Type**

bool

**Default value**

0x0

**Description**

Implement tag access permission (FEAT\_MTE\_PERM).

**has\_mte\_tag\_related\_fault\_high\_prio\_than\_data****Type**

bool

**Default value**

0x0

**Description**

For DC GZVA, Whether MMU faults generated by tag access has higher priority than faults due to data access.

**has\_nested\_virtualization****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.3 nested virtualization (FEAT\_NV). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.3 is enabled. - 2, feature is implemented.

**has\_nmi****Type**

int

**Default value**

0x1

**Description**

Implement AARCH64 Non-Maskable Interrupts (FEAT\_NMI) Possible values of this parameter are: - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_no\_os\_double\_lock****Type**

int

**Default value**

0x0

**Description**

Do not implement the OS double-lock (FEAT\_DoubleLock). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_non\_context\_synchronizing\_exception\_controls****Type**

int

**Default value**

0x1

**Description**

Implement cosmetic controls for whether exception entry and exit are context synchronizing events (SCTLR\_ELx.{EIS,EOS}) from ARMv8.5 (FEAT\_ExS). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_nv1\_raz****Type**

int

**Default value**

0x0

**Description**

Whether the implementation treats HCR\_EL2.NV1 as Read-As-Zero (RAZ), if has\_e2h\_rao = 1. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.3 is enabled. - 2, feature is implemented.

**has\_nv\_frac****Type**

int

**Default value**

0x0

**Description**

Whether the NV\_frac behavior is supported. (DEPRECATED: use nv\_frac\_support\_level)  
Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_occmo****Type**

int

**Default value**

0x1

**Description**

Implement The DC CIVAOC instruction (FEAT\_OCCMO) Possible values of this parameter are: - 1, feature is implemented if ARMv9.6 is enabled. - 2, feature is implemented.

**has\_par\_bit10\_razwi****Type**

bool

**Default value**

0x0

**Description**

Whether PAR\_EL1[10] is RAZ/WI.

**has\_partial\_delayed\_mdscr\_el1****Type**

bool

**Default value**

0x0

**Description**

has\_delayed\_oslar\_el1 only apply to some bits of MDSCR\_EL1 (MDE, KDE, TDCC, SS).

**has\_pc\_sample\_based\_profiling****Type**

bool

**Default value**

0x1

**Description**

If true, pc sample-based profiling is enabled (FEAT\_PCSRv8, FEAT\_PCSRv8p2).

**has\_pc\_sample\_profiling\_enable****Type**

bool

**Default value**

0x0

**Description**

Whether PC Sample profiling enable is implemented (FEAT\_PCSRv8p9).

**has\_pcdphint****Type**

int

**Default value**

0x0



**Description**

Support for producer-consumer data placement hints instructions (FEAT\_PCDPHINT)

Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.6 is enabled. - 2, feature is implemented.

**has\_per\_cluster\_debug\_auth\_ports****Type**

bool

**Default value**

0x0

**Description**

If true then the debug authentication ports i.e. spniden, niden, rpliden, rtpiden, dbgen, spiden are available per cluster.

**has\_permission\_indirection\_s1****Type**

int

**Default value**

0x0

**Description**

Implement the Permission Indirection Extension at stage 1 (FEAT\_S1PIE) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_permission\_indirection\_s2****Type**

int

**Default value**

0x0

**Description**

Implement the Permission Indirection Extension at stage 2 (FEAT\_S2PIE) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_permission\_overlay\_s1****Type**

int

**Default value**

0x0

**Description**

Implement the Permission Overlay Extension at stage 1 (FEAT\_S1POE) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_permission\_overlay\_s2****Type**

int

**Default value**

0x0

**Description**

Implement the Permission Overlay Extension at stage 2 (FEAT\_S2POE) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_pmss****Type**

int

**Default value**

0x0

**Description**

Implement PMU Snapshot Extension from Armv8.9 (FEAT\_PMUv3\_SS). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_pmu****Type**

int

**Default value**

0x1

**Description**

Implement the optional Performance Monitors Extension (FEAT\_PMUv3). 0, Not Implemented. 1, Implemented. 2, PMU is IMPLEMENTATION\_DEFINED, PMU version would be set to 0xF and would behave as if no PMU is implemented.

**has\_pmu\_edge\_detection****Type**

int

**Default value**

0x0

**Description**

Implement PMU Event edge detection (FEAT\_PMUv3\_EDGE) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_pmu\_extpmn****Type**

int

**Default value**

0x0

**Description**

Implement optional PMU extension feature to reserve event counters for external agents from ARMv9.5 (FEAT\_PMUv3\_EXTPMN). 0 not implemented, 1 implemented Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_pmu\_for\_sme\_extension****Type**

int

**Default value**

0x0

**Description**

Implement PMUv3 for Scalable Matrix Extension (SME) from ARMv9.5 (FEAT\_PMUv3\_SME) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_pmu\_threshold\_linking\_control****Type**

int

**Default value**

0x0

**Description**

Implement PMU threshold linking control (FEAT\_PMUv3\_TH2) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_pointer\_authentication****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.3 pointer authentication (FEAT\_PAuth). Possible values of this parameter are: - 1, feature is implemented if ARMv8.3 is enabled. - 2, feature is implemented.

**has\_prediction\_invalidation\_instructions****Type**

int

**Default value**

0x1

**Description**

Implement execution and data prediction invalidation from ARMv8.5 (FEAT\_SPECRES). Possible values of this parameter are: - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_prfm\_slc****Type**

int

**Default value**

0x0

**Description**

Implement PRFM with SLC hint (FEAT\_PRFM\_SLC). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_pstate\_dit****Type**

int

**Default value**

0x1

**Description**

Implement timing insensitivity of data processing instructions from ARMv8.4 (FEAT\_DIT). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_pstate\_pan****Type**

int

**Default value**

0x1

**Description**

Implement the PSTATE.PAN (Privileged Access Never) control from ARMv8.1 (FEAT\_PAN). Possible values of this parameter are: - 1, feature is implemented if ARMv8.1 is enabled. - 2, feature is implemented.

**has\_pstate\_uao****Type**

int

**Default value**

0x1

**Description**

Implement the PSTATE.UAO (User Access Override) control from ARMv8.2 (FEAT\_UAO). Possible values of this parameter are: - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_qarma3\_pac****Type**

bool

**Default value**

0x0

**Description**

Supports QARMA3 pointer authentication algorithm (FEAT\_PACQARMA3).

**has\_ras****Type**

int

**Default value**

0x0

**Description**

Implements the ARMv8 RAS Extension. 0 = NO\_RAS, 1 = MINIMAL\_RAS, 2 = FULL\_RAS (FEAT\_RAS).

**has\_ras\_aderr****Type**

int

**Default value**

0x0

**Description**

Implement RAS Asynchronous Device Read Error from Armv8.9 (FEAT\_ADERR). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_ras\_anerr****Type**

int

**Default value**

0x0

**Description**

Implement RAS Asynchronous Normal Read Error from Armv8.9 (FEAT\_ANERR). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_ras\_armv84\_extension****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.4 RAS Extension (FEAT\_RASv1p1). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_ras\_armv89\_double\_fault****Type**

int

**Default value**

0x0

**Description**

Implement RAS Double Fault Extension from Armv8.9 (FEAT\_DoubleFault2). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_ras\_armv89\_extension****Type**

int

**Default value**

0x1

**Description**

Implement RAS extension from Armv8.9 (FEAT\_RASv2). Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_ras\_critical\_error****Type**

int

**Default value**

0x0

**Description**

[DEPRECATED: Set CI field on first register in error\_record\_feature\_register JSON instead]  
ARMv8.4 AArch64 RAS Critical Error is implemented or not. 0 - Feature Not Supported, 1 - Feature always enabled, 2 - Feature is controllable.

**has\_ras\_delegated\_serror\_exceptions\_for\_el3****Type**

int

**Default value**

0x1

**Description**

Implement Delegated SError exceptions for EL3 (FEAT\_E3DSE). Possible values of this parameter are: - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_ras\_double\_fault****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.4 RAS Double Fault Extension (FEAT\_DoubleFault). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_ras\_fault\_injection****Type**

int

**Default value**

0x0

**Description**

[DEPRECATED: Set INJ field on first register in error\_record\_feature\_register JSON instead]  
Implement ARMv8.4 Standard Fault Injection mechanism. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_ras\_mmmap\_view****Type**

bool

**Default value**

0x0

**Description**

Implement memory mapped view of RAS Registers for cores.

**has\_ras\_pfar****Type**

int

**Default value**

0x0

**Description**

Implement RAS Physical Fault Address Registers from Armv8.9 (FEAT\_PFAR). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_ras\_timestamp****Type**

int

**Default value**

0x0

**Description**

[DEPRECATED: Set TS field on first register in error\_record\_feature\_register JSON instead]  
ARMv8.4 AArch64 RAS Timestamp register is implemented or not. 0 - No Timestamp is recorded, 1 - Generic Timer timestamp is recorded, 2 - IMP DEF timestamp is recorded.

**has\_rassa\_acr****Type**

bool

**Default value**

0x0

**Description**

Implement RAS System Architecture v2 optional access control register (FEAT\_RASSA\_ACR).

**has\_restriction\_on\_speculative\_data\_loaded****Type**

int



**Default value**

0x1

**Description**

Implements the ARMv8.5 security feature (Restrictions on the effects of speculation) (FEAT\_CSV3). Possible values of this parameter are: - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_rme****Type**

bool

**Default value**

0x0

**Description**

If true, implements full realm management extension (FEAT\_RME). Note: This parameter is deprecated and will be removed in future releases, please use rme\_support\_level parameter.

**has\_rme\_gdi****Type**

int

**Default value**

0x0

**Description**

Support for RME granular data isolation (FEAT\_RME\_GDI)) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.6 is enabled. - 2, feature is implemented.

**has\_rme\_gpc2****Type**

bool

**Default value**

0x0

**Description**

If true then RME GPC2 extension is enabled (FEAT\_RME\_GPC2).

**has\_rme\_gpc3****Type**

bool

**Default value**

0x0

**Description**

If true then RME GPC3 extension is enabled (FEAT\_RME\_GPC3).

**has\_rndr****Type**

int

**Default value**

0x0

**Description**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_rndr\_trap****Type**

int

**Default value**

0x0

**Description**

Implement trapping for RNDR and RNDRSS random number registers from ARMv8.8. (FEAT\_RNG\_TRAP) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_rounding\_doubling\_multiply\_add\_subtract****Type**

int

**Default value**

0x1

**Description**

Implement the rounding doubling multiply add and subtract instructions from ARMv8.1 (FEAT\_RDM). Possible values of this parameter are: - 1, feature is implemented if ARMv8.1 is enabled. - 2, feature is implemented.

**has\_rprfm****Type**

bool

**Default value**

0x0

**Description**

Support for RPRFM hint instruction (FEAT\_RPRFM).

**has\_sbistc\_ae25****Type**

bool

**Default value**

0x0

**Description**

Implements the SBISTC for AE25 class cpu.

**has\_sctlr2****Type**

int

**Default value**

0x1

**Description**

Implement SCTL2\_ELx registers (FEAT\_SCTL2) Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_sebep****Type**

int

**Default value**

0x0

**Description**

Implement Synchronous-Exception-Based Event Profiling from ARMv9.4 (FEAT\_SEBEP). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.4 is enabled. - 2, feature is implemented.

**has\_secure\_el2****Type**

int

**Default value**

0x1

**Description**

Implement support for Secure EL2 (FEAT\_SEL2). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_self\_hosted\_trace\_extension****Type**

int

**Default value**

0x1

**Description**

Implement support for the Self-hosted Trace Extensions from ARMv8.4 (FEAT\_TRF). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_small\_page\_table****Type**

int

**Default value**

0x1

**Description**

Implement small page table support which increases the maximum value of TxSZ field from ARMv8.4 (FEAT\_TTST). Note: will be unimplemented only if both has\_small\_page\_table=0x0 and has\_secure\_el2=0x0. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_software\_lock****Type**

bool

**Default value**

0x1

**Description**

Implement software lock in memory-mapped CTI, PMU, and external debug interfaces.

**has\_spe\_eft****Type**

int

**Default value**

0x1

**Description**

Implement SPE extended operation type filtering from ARMv9.5 (FEAT\_SPE\_EFT) Possible values of this parameter are: - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_spe\_fds****Type**

int

**Default value**

0x0

**Description**

Implement SPE filter by data source from ARMv8.9 (FEAT\_SPE\_FDS) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_spe\_for\_sme\_extension****Type**

int

**Default value**

0x1

**Description**

Implement support of SME to SPE from ARMv9.5 (FEAT\_SPE\_SME). Possible values of this parameter are: - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_spe\_fpf****Type**

int

**Default value**

0x1

**Description**

Implement SPE operation type extension for ASIMD and FP from ARMv9.5 (FEAT\_SPE\_FPF) Possible values of this parameter are: - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_spe\_nvm****Type**

int

**Default value**

0x1

**Description**

Implement Statistical Profiling physical address mode (FEAT\_SPE\_nVM) Possible values of this parameter are: - 1, feature is implemented if ARMv9.6 is enabled. - 2, feature is implemented.

**has\_speculation\_barrier\_inst****Type**

int

**Default value**

0x1

**Description**

Implement speculation barrier instruction (SB) from ARMv8.5 (FEAT\_SB). Possible values of this parameter are: - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_speculative\_sei****Type**

bool

**Default value**

0x0

**Description**

If true, the PE can generate SError interrupt exceptions from speculative reads of memory, including speculative instruction fetches.

**has\_srmask****Type**

int

**Default value**

0x1

**Description**

Implement bitwise write masks for EL1 control registers (FEAT\_SRMASK). Possible values of this parameter are: - 1, feature is implemented if ARMv9.6 is enabled. - 2, feature is implemented.

**has\_stage2\_ap\_speculative\_update****Type**

int

**Default value**

0x0

**Description**

Speculative update of S2 AP bit on S1 TTW. 0 = No Update, 1 = Update, 2 = Update including for AT ops.

**has\_stage2\_fwb****Type**

int

**Default value**

0x1

**Description**

Implement HCR\_EL2.FWB, stage 2 control of memory types and cacheability (FEAT\_S2FWB). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_stage2\_xnx****Type**

int

**Default value**

0x1

**Description**

Implement the extended XN[1:0] stage 2 control from ARMv8.2 (FEAT\_XNX). Possible values of this parameter are: - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_stage2\_xnx\_in\_aarch32****Type**

bool

**Default value**

0x1

**Description**

Implement the extended XN[1:0] stage 2 control from ARMv8.2 for Aarch32 (FEAT\_XNX).

**has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**has\_supersections****Type**

bool

**Default value**

0x1

**Description**

Whether VMSAv8-32 supersection to support more than 32-bit PA using short descriptor is implemented.

**has\_sve****Type**

int

**Default value**

0x0

**Description**

Whether SVE is implemented (FEAT\_SVE). Note: this is required to enable SME (FEAT\_SME) with sve.has\_sme=1. An SME only implementation can be enabled by setting both as well as sve.sme\_only=1.

**has\_synchronous\_load\_atomics****Type**

bool

**Default value**

0x1

**Description**

Report asynchronous abort due to unsupported load atomics as synchronous (Cacheable).

**has\_synchronous\_load\_atomics\_noncacheable****Type**

bool

**Default value**

0x1

**Description**

Report asynchronous abort due to unsupported load atomics as synchronous (Non-Cacheable).

**has\_synchronous\_store\_atomics****Type**

bool

**Default value**

0x0

**Description**

Report asynchronous abort due to unsupported store atomics as synchronous (Cacheable).

**has\_synchronous\_store\_atomics\_noncacheable****Type**

bool

**Default value**

0x0



**Description**

Report asynchronous abort due to unsupported store atomics as synchronous (Non-Cacheable).

**has\_sysinstr128****Type**

int

**Default value**

0x0

**Description**

Support for System Instructions that can take 128-bit inputs (FEAT\_SYSINSTR128) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.4 is enabled. - 2, feature is implemented.

**has\_sysreg128****Type**

int

**Default value**

0x0

**Description**

Support for 128-bit System Registers (FEAT\_SYSREG128) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.4 is enabled. - 2, feature is implemented.

**has\_tcr2****Type**

int

**Default value**

0x1

**Description**

Implement TCR2\_ELx registers (FEAT\_TCR2) Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_tidcp1****Type**

int

**Default value**

0x1

**Description**

Implement Armv8.8 ELO use of implementation defined functionality (FEAT\_TIDCP1)

Possible values of this parameter are: - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_tlb\_conflict\_abort****Type**

bool

**Default value**

0x0

**Description**

Detected inconsistent TLB content generate aborts.

**has\_tlb\_pa\_caching****Type**

bool

**Default value**

0x0

**Description**

Whether intermediate caching of translation table walks might include NonCoherent caches of previous valid walks. 0, NonCoherent caches might be included. 1, No NonCoherent caches included (FEAT\_nTLBPA).

**has\_tlbi\_range****Type**

int

**Default value**

0x1

**Description**

Implement support for TLB Range Maintenance instructions (TLBI RVAE1, etc) from ARMv8.4 (FEAT\_TLBI RANGE). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_tlbi\_to\_outer\_shareable****Type**

int

**Default value**

0x1

**Description**

Implement support for TLB Maintenance instructions that extend to the Outer Shareable domain (TLBI VAE1OS, etc) from ARMv8.4 (FEAT\_TLBIOS). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_tlbi\_ttl****Type**

int

**Default value**

0x1

**Description**

Implement support for the TTL level hint in by-address TLB Maintenance instructions from ARMv8.4 (FEAT\_TTL). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_tlbiw****Type**

int

**Default value**

0x0

**Description**

Implement TLBI instruction for stage2 dirty (FEAT\_TLBIW). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.5 is enabled. - 2, feature is implemented.

**has\_tme****Type**

bool

**Default value**

0x0

**Description**

If true, implements TME, the Transactional Memory Extension (FEAT\_TME).

**has\_translation\_hardenening****Type**

int

**Default value**

0x0

**Description**

Implement the Translation Hardening Extension (FEAT\_THE). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_trbe****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Trace Buffer Extension (FEAT\_TRBE).

**has\_trbe\_ext****Type**

int

**Default value**

0x0

**Description**

Implements the Trace Buffer external mode extension (FEAT\_TRBE\_EXT). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv9.4 is enabled. - 2, feature is implemented.

**has\_uinj****Type**

int

**Default value**

0x1

**Description**

Implement software injection of Undefined Instruction exceptions (FEAT\_UINJ). Possible values of this parameter are: - 1, feature is implemented if ARMv9.6 is enabled. - 2, feature is implemented.

**has\_unaligned\_single\_copy\_atomicity****Type**

int

**Default value**

0x1

**Description**

Implement support for SCTLR\_ELx.nAA from ARMv8.4, and A64 atomic, exclusive and acquire/release instructions accessing unaligned bytes inside a 16byte window will not

generate alignment fault (FEAT\_LSE2). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

### **has\_unsupported\_exclusive\_fault**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

Report unsupported exclusive access with Unsupported Exclusive fault status (otherwise use external abort).

### **has\_v8\_4\_debug\_extension**

#### **Type**

int

#### **Default value**

0x1

#### **Description**

Implement ARMv8.4 debug extensions (FEAT\_Debugv8p4). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

### **has\_v8\_4\_pmu\_extension**

#### **Type**

int

#### **Default value**

0x1

#### **Description**

Implement PMU extension from ARMv8.4 (FEAT\_PMUv3p4). Possible values of this parameter are: - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

### **has\_v8\_5\_debug\_over\_power\_down**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

Implement ARMv8.5 Debug over powerdown (FEAT\_DoPD). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_v8\_5\_pmu\_extension****Type**

int

**Default value**

0x1

**Description**

Implement PMU extension from ARMv8.5 (FEAT\_PMUv3p5). Possible values of this parameter are: - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_v8\_5\_spe\_extension****Type**

int

**Default value**

0x1

**Description**

Implement SPE extension from ARMv8.5 (FEAT\_SPEv1p1). Possible values of this parameter are: - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_v8\_6\_pmu\_events****Type**

int

**Default value**

0x1

**Description**

Implements PMU events from ARMv8.6 Possible values of this parameter are: - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**has\_v8\_7\_fp\_enhancements****Type**

int

**Default value**

0x1

**Description**

Implements the Floating Point enhancements from Armv8.7 (introduces FPCR.FIZ/AH/NEP, etc. (FEAT\_AFP).) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_v8\_7\_pmu\_events****Type**

int

**Default value**

0x1

**Description**

Implement PMU events from ARMv8.7. Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_v8\_7\_pmu\_extension****Type**

int

**Default value**

0x1

**Description**

Implement PMU extension from ARMv8.7 (FEAT\_PMUv3p7). Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_v8\_7\_spe\_extension****Type**

int

**Default value**

0x1

**Description**

Implement SPE extension from ARMv8.7 (FEAT\_SPEv1p2) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_v8\_7\_spe\_inverted\_filtering****Type**

bool

**Default value**

0x1

**Description**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

**has\_v8\_7\_spe\_previous\_branch\_target****Type**

bool

**Default value**

0x1

**Description**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

**has\_v8\_8\_debug\_extension****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.8 debug extensions (FEAT\_Debugv8p8). Possible values of this parameter are: - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_v8\_8\_pmu\_events****Type**

int

**Default value**

0x1

**Description**

Implement PMU events from ARMv8.8 (FEAT\_PMUv3). Possible values of this parameter are: - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_v8\_8\_pmu\_extension****Type**

int

**Default value**

0x1

**Description**

Implement PMU extension from ARMv8.8 (FEAT\_PMUv3p8). Possible values of this parameter are: - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_v8\_8\_spe\_extension****Type**

int

**Default value**

0x1

**Description**

Implement SPE extension from ARMv8.8 (FEAT\_SPEv1p3). Possible values of this parameter are: - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.



**has\_v8\_9\_debug\_extension****Type**

int

**Default value**

0x1

**Description**

Implement ARMv8.9 debug extensions (FEAT\_Debugv8p9) Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_v8\_9\_pc\_sample\_based\_profiling****Type**

int

**Default value**

0x1

**Description**

Implement PC Sample-based Profiling from ARMv8.9 (FEAT\_PCSRv8p9) Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_v8\_9\_pmu\_events****Type**

int

**Default value**

0x1

**Description**

Implement PMU events from ARMv8.9 (FEAT\_PMUv3). Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_v8\_9\_pmu\_extension****Type**

int

**Default value**

0x1

**Description**

Implement PMU extension from ARMv8.9 (FEAT\_PMUv3p9). Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_v8\_9\_spe\_extension****Type**

int

**Default value**

0x1

**Description**

Implement SPE extension from ARMv8.9 (FEAT\_SPEv1p4) Possible values of this parameter are: - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**has\_v9\_6\_spe\_extension****Type**

int

**Default value**

0x1

**Description**

Implement FEAT\_SPEv1p5 and FEAT\_SPE\_EXC from ARMv9.6. Possible values of this parameter are: - 1, feature is implemented if ARMv9.6 is enabled. - 2, feature is implemented.

**has\_vnocr\_el2****Type**

int

**Default value**

0x1

**Description**

Implement support for nested virtualization enhancements from ARMv8.4 (FEAT\_NV2). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**has\_wfet\_and\_wfit****Type**

int

**Default value**

0x1

**Description**

Implements WFE and WFI with Timeout from Armv8.7 (FEAT\_WFXT) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_writebuffer****Type**

bool

**Default value**

0x0

**Description**

Implement write accesses buffering before L1 cache. May affect ext\_abort behaviour.

**has\_xs****Type**

int

**Default value**

0x1

**Description**

Implements Armv8.7 XS, TLBInXS, DSBnXS instruction (FEAT\_XS). Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**hcptr\_tta\_behaviour****Type**

int

**Default value**

0x2

**Description**

Behaviour of HCPTR.TTA when there is no CP14 ETM interface. 0, RAZ/WI. 1, RAO/WI. 2, stateful.

**hcr\_el2\_miocnce\_is\_rw****Type**

bool

**Default value**

0x0

**Description**

If true, HCR\_EL2.MIOCNCE is treated as R/W instead of RAZ/WI.

**hcr\_swio\_res1****Type**

bool

**Default value**

0x0

**Description**

Whether HCR.SWIO and/or HCR\_EL2.SWIO are RES1.

**hdbss\_error\_fault\_type****Type**

int

**Default value**

0x0

**Description**

Type of fault reported for HDBSS errors. 0 = precise exception, 1 = fault logged in HDBSSPROD\_EL2.FSC (FEAT\_HDBSS).

**hpfar\_unknown\_when\_ipa\_invalid****Type**

bool

**Default value**

0x0

**Description**

If true, HPFAR\_EL2 is set to 0 when IPA is not valid for stage 2 faults.

**hpfar\_update\_behaviour****Type**

int

**Default value**

0x0

**Description**

Defines HPFAR\_EL2 update condition.0: Always updated on faults taken to EL2.1: Only when IPA is valid.2: When IPA is valid or unknown.

**hsr\_uncond\_cc****Type**

bool

**Default value**

0x0

**Description**

Condition codes reported in HSR as AL if it passes.

**hvbar\_reset\_is\_rvbar****Type**

bool

**Default value**

0x0

**Description**

If true then the reset value of HVBAR is RVBAR, if false the reset value is UNKNOWN.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-log2linelen****Type**

int

**Default value**

0x0

**Description**

If nonzero, Log2 of the instruction cache line length in bytes (valid values in range 4-8). Otherwise the value of cache-log2linelen is used.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-nprefetch****Type**

int

**Default value**

0x1

**Description**

Number of next sequential instruction cache lines to prefetch. This is only used when `icache-prefetch_enabled=true`.

**`icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-prefetch_level`****Type**

int

**Default value**

0x0

**Description**

0 based cache level at which instructions are pre-fetched. This is only used when `icache-prefetch_enabled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_bus_width_in_bytes`****Type**

int

**Default value**

0x8

**Description**

L1 I-Cache read bus width in bytes used to calculate per-access timing annotations.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**icache-ways****Type**

int

**Default value**

0x2

**Description**

L1 I-Cache number of ways (sets are implicit from size).

**id\_spec\_fpacc\_raz****Type**

bool

**Default value**

0x0

**Description**

If true, implementation opts not to disclose the speculative use of pointers processed by a PAC authentication failure by having value 0 for Spec\_FPACC bits of ID\_AA64MMFR3\_EL1 register.

**idte3\_support\_level****Type**

int

**Default value**

0x0

**Description**

Support for trapping ID register accesses to EL3(FEAT\_IDTE3): 0 - Not implemented. 1 - Implemented.

**ignore\_DBGPRCR\_CWRR****Type**

bool

**Default value**

0x0

**Description**

Ignore writes to the deprecated DBGPRCR.CWRR bit.

**ignore\_access\_flag\_update\_by\_CMOs****Type**

bool

**Default value**

0x0

**Description**

If true, CMOs(cache maintenance operations) neither update the access flag nor generate access flag fault.

**ignore\_access\_flag\_update\_by\_at\_ops****Type**

bool



**Default value**

0x0

**Description**

If true, AT operations do not update access flag.

**`ignore_data_abt_on_af_update_by_at_ops`****Type**

bool

**Default value**

0x1

**Description**

If true, Data abort generated on AF update by AT operations are ignored. This parameter is only valid if `ignore_access_flag_update_by_at_ops` is false.

**`ignore_large_address_top_bits_in_page_walk`****Type**

bool

**Default value**

0x0

**Description**

Whether page table bits [15:12] are ignored if `PA_SIZE < 52` and output address is configured `< 52` with large page.

**`ignore_tag_check_dcc_load_store_in_ma_mode_when_tco_is_disabled`****Type**

bool

**Default value**

0x0

**Description**

Constrained unpredictable behavior for reads/writes to external debug interface DTR regs in memory access mode when `PSTATE.TCO` is 0. If true, tag check is ignored else, tag check is performed if required.

**`ignore_traps_to_dcc_regs_in_debug`****Type**

bool

**Default value**

0x0

**Description**

Whether traps get ignored for the following registers in debug state: \* AArch64: MDCCSR\_ELO, OSDTREX\_EL1, OSDTRTX\_EL1, MDCCINT\_EL1. \* AArch32: DBGDSCRint, DBGDIDR, DBGDSAR, DBGDRAR, DBGDTRRXext, DBGDTRTXext, DBGDCCINT.

**illegal\_state\_exception\_priority****Type**

int

**Default value**

0x0

**Description**

IMPDEF priority of Illegal State Exception. 0: After breakpoint exceptions 1: Before Instruction Abort.

**imp\_def\_functionality\_behaviour****Type**

int

**Default value**

0x0

**Description**

Behaviour of IMPLEMENTATION DEFINED registers and system instructions. 0, UNDEF. 1, RAZ/WI.

**impdef\_regs\_and\_unpred\_from\_implementation****Type**

string

**Default value**

""

**Description**

Configure implementation defined registers and unpredictable behaviour to match the specified implementation. Requires a license for the selected implementation model. User has to provide the default values for the published or configurable parameters through commandline arguments. Use ARM\_Cortex-A<num> or ARM\_<codename> for licensed pre-release cores.

**impdef\_sysreg\_json****Type**

string

**Default value**

"[]"

**Description**

Configure mask/reset bitmasks for impdef. registers in a JSON format which is (where 'bitwise' indicates 'reset'/'mask' to flip existing bits): [{ "name": "IMP\_SYSREGO\_EL1", "reset":0, "mask":0, "encoding":0, "bitwise":true}\* ].

**independent\_cache\_control\_traps****Type**

int

**Default value**

0x0

**Description**

Implement Independent Cache Control traps from ARMv8.5. 0, No support. 1, Supported but not for tlb maintenance instructions. 2, Full support. (FEAT\_EVT).

**insert\_iesb\_before\_exception****Type**

bool

**Default value**

0x0

**Description**

If true then inserts an IESB before taking with Exception otherwise has no effect and IESB is taken after PState is changed due to the Exception.

**instruction\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**internal\_vgic****Type**

bool

**Default value**

0x0

**Description**

Instantiate VGIC peripheral in this processor.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**is\_debug\_state\_pmu\_snapshot\_allowed****Type**

bool

**Default value**

0x1

**Description**

If true, PMU snapshot is allowed in debug state.

**is\_first\_pcsr\_sample\_ignored****Type**

bool

**Default value**

0x0

**Description**

If true, First read of PMPCSR register after reset returns 0xFFFFFFFF.

**is\_mt\_res0****Type**

bool

**Default value**

0x0

**Description**

If ARMv8.6 is not implemented, and PMUV3 is implemented, this parameter controls whether PMEVTYPER<n>.MT bit is RES0 or RW. For other implementations, this parameter has no effect.

**is\_serror\_edge\_triggered****Type**

bool

**Default value**

0x1

**Description**

If true, SError is edge-triggered. Otherwise, its level-triggered.

**`is_tagged_nsh_treated_as_tagged`****Type**

bool

**Default value**

0x1

**Description**

Whether a tagged NonShared memory attribute is treated as tagged or not.

**`is_uniprocessor`****Type**

bool

**Default value**

0x0

**Description**

Value for the U bit in MPIDR. true disables L1 cache coherency protocols.

**`isb_is_branch`****Type**

bool

**Default value**

0x0

**Description**

If true, ISB is considered an immediate branch. This allows to count ISB as a branch in BRBE.

**`ish_is_osh`****Type**

bool

**Default value**

0x0

**Description**

Whether Innershareable is same as OuterShareable.

**`itd_conditional_instructions_are_32bit`****Type**

bool

**Default value**

0x0

**Description**

When SCTLR\_ELx.ITD=1, an IT instruction plus a T16 instruction are considered a single 32bit conditional instruction.

**`jidr_is_undef_at_el0`****Type**

bool

**Default value**

0x0

**Description**

If true, JIDR register access is UNDEF at EL0.

**`jmcrr_is_undef_at_el0`****Type**

bool

**Default value**

0x0

**Description**

If true, JMCRR register access is UNDEF at EL0.

**`joscrr_is_undef_at_el0`****Type**

bool

**Default value**

0x0

**Description**

If true, JOSCR register access is UNDEF at EL0.

**`l2cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**`l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`l2cache-read_bus_width_in_bytes`****Type**

int

**Default value**

0x8

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**`l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l2cache-ways`****Type**

int

**Default value**

0x10

**Description**

L2 Cache number of ways (sets are implicit from size).



**l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x8

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x0

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l3cache-mpamf.arch\_major\_ver****Type**

int

**Default value**

0x0

**Description**

L3 Cache MPAMF\_AIDR architecture major version.

**l3cache-mpamf.arch\_minor\_ver****Type**

int

**Default value**

0x0

**Description**

L3 Cache MPAMF\_AIDR architecture minor version.

**l3cache-mpamf.bwa\_width\_ns****Type**

int

**Default value**

0x10

**Description**

L3 Cache width of MPAM bandwidth allocation fields for non-secure accesses.

**l3cache-mpamf.bwa\_width\_s****Type**

int

**Default value**

0x10

**Description**

L3 Cache width of MPAM bandwidth allocation fields for secure accesses.

**l3cache-mpamf.cmax\_width\_ns****Type**

int

**Default value**

0x0

**Description**

L3 Cache number of fractional bits in MPAM cache capacity partition control for non-secure accesses. Only the register interface is implemented - the control is NOT FUNCTIONAL.

**l3cache-mpamf.cmax\_width\_s****Type**

int

**Default value**

0x0

**Description**

L3 Cache number of fractional bits in MPAM cache capacity partition control for secure accesses. Only the register interface is implemented - the control is NOT FUNCTIONAL.

**l3cache-mpamf.cpbm\_width\_ns****Type**

int

**Default value**

0x0

**Description**

L3 Cache width of MPAM cache portion bitmap for non-secure accesses. If 0, the feature is not implemented, and all ways are available.

**13cache-mpamf.cpbm\_width\_rl****Type**

int

**Default value**

0x0

**Description**

L3 Cache width of MPAM cache portion bitmap for realm accesses. If 0, the feature is not implemented, and all ways are available.

**13cache-mpamf.cpbm\_width\_rt****Type**

int

**Default value**

0x0

**Description**

L3 Cache width of MPAM cache portion bitmap for root accesses. If 0, the feature is not implemented, and all ways are available.

**13cache-mpamf.cpbm\_width\_s****Type**

int

**Default value**

0x0

**Description**

L3 Cache width of MPAM cache portion bitmap for secure accesses. If 0, the feature is not implemented, and all ways are available.

**13cache-mpamf.csu\_num\_mon\_ns****Type**

int

**Default value**

0x0

**Description**

L3 Cache number of MPAM cache storage usage monitors for non-secure accesses.

**l3cache-mpamf.csu\_num\_mon\_rl****Type**

int

**Default value**

0x0

**Description**

L3 Cache number of MPAM cache storage usage monitors for realm accesses.

**l3cache-mpamf.csu\_num\_mon\_rt****Type**

int

**Default value**

0x0

**Description**

L3 Cache number of MPAM cache storage usage monitors for root accesses.

**l3cache-mpamf.csu\_num\_mon\_s****Type**

int

**Default value**

0x0

**Description**

L3 Cache number of MPAM cache storage usage monitors for secure accesses.

**l3cache-mpamf.esr\_mask****Type**

int

**Default value**

0xffffffff

**Description**

L3 Cache MPAMF\_ESR mask value.

**l3cache-mpamf.has\_esr****Type**

bool

**Default value**

0x0

**Description**

L3 Cache's MPAMF\_ESR, MPAMF\_ECR, and MPAM error handling implemented.

**l3cache-mpamf.has\_extd\_esr****Type**

bool

**Default value**

0x0

**Description**

L3 Cache's MPAMF\_ESR is 64-bits.

**l3cache-mpamf.has\_impl\_idr****Type**

bool

**Default value**

0x0

**Description**

L3 Cache's MPAMF\_IMPL\_IDR is present.

**l3cache-mpamf.has\_mbwu\_long\_counter****Type**

bool

**Default value**

0x0

**Description**

L3 Cache has long MBWU counter and capture registers.

**l3cache-mpamf.has\_mpamfidr\_ext****Type**

bool

**Default value**

0x0

**Description**

MPAMF\_IDR.EXT support.

**l3cache-mpamf.has\_partid\_nrw****Type**

bool

**Default value**

0x0

**Description**

Narrowing part ID register is present. This is global rather than per-instance.

**l3cache-mpamf.has\_priority\_partitioning****Type**

bool

**Default value**

0x0

**Description**

The selected resource has priority partitioning described in MPAMF\_PRI\_IDR.

**l3cache-mpamf.has\_prod\_id****Type**

int

**Default value**

0x0

**Description**

L3 Cache MPAMF\_IIDR product ID supported.

**l3cache-mpamf.has\_prod\_rev****Type**

int

**Default value**

0x0

**Description**

L3 Cache MPAMF\_IIDR product REVISION supported.

**l3cache-mpamf.has\_prod\_var****Type**

int

**Default value**

0x0

**Description**

L3 Cache MPAMF\_IIDR product VARIANT supported.

**l3cache-mpamf.has\_prop\_ns****Type**

bool

**Default value**

0x0

**Description**

Enable memory bandwidth proportional stride control for non-secure accesses. Only the register interface is implemented - the control is NOT FUNCTIONAL.

**l3cache-mpamf.has\_prop\_s****Type**

bool

**Default value**

0x0

**Description**

Enable memory bandwidth proportional stride control for secure accesses. Only the register interface is implemented - the control is NOT FUNCTIONAL.

**l3cache-mpamf.has\_ris****Type**

bool

**Default value**

0x0

**Description**

L3 Cache has resource instance selection support.

**l3cache-mpamf.max\_partid\_ns****Type**

int

**Default value**

0xffff

**Description**

L3 Cache Maximum value of non-secure PARTID supported.

**l3cache-mpamf.max\_partid\_rl****Type**

int

**Default value**

0xffff

**Description**

L3 Cache Maximum value of realm PARTID supported for RME implementations.

**l3cache-mpamf.max\_partid\_rt****Type**

int



**Default value**`0xffff`**Description**

L3 Cache Maximum value of root PARTID supported for RME implementations.

**`l3cache-mpamf.max_partid_s`****Type**`int`**Default value**`0xffff`**Description**

L3 Cache Maximum value of secure PARTID supported.

**`l3cache-mpamf.max_pmg_ns`****Type**`int`**Default value**`0xff`**Description**

L3 Cache Maximum value of non-secure PMG supported.

**`l3cache-mpamf.max_pmg_r1`****Type**`int`**Default value**`0xff`**Description**

L3 Cache Maximum value of realm PMG supported for RME implementations.

**`l3cache-mpamf.max_pmg_rt`****Type**`int`**Default value**`0xff`**Description**

L3 Cache Maximum value of root PMG supported for RME implementations.

**`l3cache-mpamf.max_pmg_s`****Type**`int`

**Default value**`0xff`**Description**

L3 Cache Maximum value of secure PMG supported.

**`l3cache-mpamf.mbwu_long_counter_width`****Type**`int`**Default value**`0x0`**Description**

L3 Cache long MBWU counter width. 0: 63 bits, 1: 44 bits.

**`l3cache-mpamf.no_impl_msmon`****Type**`bool`**Default value**`0x0`**Description**

L3 Cache's MPAMF\_IMPL\_IDR does not describe resource monitors.

**`l3cache-mpamf.no_impl_part`****Type**`bool`**Default value**`0x0`**Description**

L3 Cache's MPAMF\_IMPL\_IDR does not describe resource partitioning controls.

**`l3cache-mpamf.ris_max`****Type**`int`**Default value**`0x0`**Description**

L3 Cache's largest resource instance selector value defined.

**`l3cache-mpamf_base`****Type**`int`

**Default value**

0x0

**Description**

L3 Cache memory mapped MPAM registers base address.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x8

**Description**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x0

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x8

**Description**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**legacy\_combining\_exc\_catch\_trace****Type**

bool

**Default value**

0x1

**Description**

Whether exception catch is traced as part of exception entry/exit in same cycle.

**log2\_trace\_buffer\_alignment****Type**

int

**Default value**

0x0

**Description**

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

**ls64\_ignore\_s1\_unpred\_memattr\_transformation****Type**

bool

**Default value**

0x0

**Description**

If true, stage 1 unpredictable memory attribute transformations are ignored for FEAT\_LS64 single-copy atomic 64-byte load/store instructions' (FEAT\_LS64, FEAT\_LS64\_V, FEAT\_LS64\_ACCDATA).

**ls64\_memtype\_check\_use\_combined\_memattr****Type**

int

**Default value**

0x0

**Description**

FEAT\_LS64 single-copy atomic 64-byte load/store instructions' 0 : memory attributes check is performed at each enabled stage of translation, 1 : memory attributes check is done on the combined memory attributes only. 2. memory attributes check is done on the combined memory attributes with Stage1 and Stage2 fault get evaluated to check on which stage fault should be reported.

**ls64wb\_memtype\_check\_allow\_any\_cacheable\_memattr****Type**

bool

**Default value**

0x0

**Description**

If true, when FEAT\_LS64WB is implemented, any cacheable memory access performed by LD/ST64B instructions is 64-byte, single-copy atomic.

**max\_32bit\_el****Type**

int

**Default value**

0x3

**Description**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, x: [0:3] - All the levels below supplied ELx supports A32.

**mdrar\_el1\_res0****Type**

bool

**Default value**

0x0

**Description**

MDRAR\_EL1 is RESO.

**mdselr\_le\_16\_bps\_wps\_behaviour****Type**

int

**Default value**

0x0

**Description**

Behaviour of MDSELR\_EL1 and related traps/enables if fewer than 16 watchpoints and fewer than 16 breakpoints are implemented: 0 - MDSELR\_EL1 is stateful; 1 - MDSELR\_EL1, EBWE, FGTS are RAZ/WI, traps and enables do not apply; 2 - MDSELR\_EL1, EBWE, FGTS with checked traps.

**mec\_support\_level****Type**

int

**Default value**

0x0

**Description**

0 -> Memory Encryption Contexts not implemented, 1 -> LEGACY\_TZ\_EN mode i.e. MEC register fields are stateful but only supports secure/non-secure states, 2 -> Memory Encryption Contexts fully implemented (FEAT\_MEC).

**memory.acp.AxCACHE\_mask****Type**

int

**Default value**

0x0

**Description**

Used with memory.acp.AxCACHE\_pattern to define which memory types the ACP port accepts. All transactions which do not satisfy (AxCACHE & mask) == pattern will abort.

**memory.acp.AxCACHE\_pattern****Type**

int

**Default value**

0x0

**Description**

Used with memory.acp.AxCACHE\_mask to define which memory types the ACP port accepts. All transactions which do not satisfy (AxCACHE & mask) == pattern will abort.

**memory.l2\_cache.is\_inner\_cacheable****Type**

bool

**Default value**

0x1

**Description**

L2 cache obeys inner cacheable attributes (rather than outer cacheable attributes).

**memory.l2\_cache.is\_inner\_shareable****Type**

bool

**Default value**

0x1

**Description**

L2 cache obeys inner shareable attributes (rather than outer shareable attributes).

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x0

**Description**

Specify the memory tagging extension support level: 0, not implemented. 1, instructions and registers only are implemented (FEAT\_MTE). 2, implemented (FEAT\_MTE2). 3, implemented with asymmetric handling of exceptions (FEAT\_MTE3). 4, implemented (FEAT\_MTE4).

**mixed\_endian****Type**

int

**Default value**

0x1

**Description**

Implement support for mixed endianness. 0, not supported. 1, supported at all exception levels. 2, supported at ELO only.

**mops\_cpy\_block\_size****Type**

int

**Default value**

0x40



**Description**

Block size used for memcpy memory accesses.

**mops\_cpy\_default\_dir****Type**

int

**Default value**

0x0

**Description**

Default direction for non-overlapping memcpy operations: 0, forwards. 1, backwards.

**mops\_cpy\_handle\_async\_exceptions****Type**

bool

**Default value**

0x0

**Description**

Handle any pending async exceptions after copying a block of data, instead of waiting until instruction end.

**mops\_cpy\_post\_size****Type**

int

**Default value**

0xa

**Description**

Percentage of data copied in memcpy 'E' instructions.

**mops\_cpy\_pre\_size****Type**

int

**Default value**

0xa

**Description**

Percentage of data copied in memcpy 'P' instructions.

**mops\_cpy\_pre\_size\_threshold****Type**

int

**Default value**

0x0

**Description**

Size threshold in Bytes for CPY\*P\* instructions.

**mops\_cpy\_single\_access****Type**

bool

**Default value**

0x0

**Description**

Execute memcpy as a single read and single write access.

**mops\_cpy\_write\_abort\_before\_read****Type**

bool

**Default value**

0x0

**Description**

Report the data aborts and watchpoint of the write accesses, before those of the read accesses.

**mops\_cpy\_zero\_size\_can\_fault****Type**

bool

**Default value**

0x1

**Description**

Fault because of mismatched implementation option when the operation is of size 0.

**mops\_exec\_order\_can\_fault****Type**

bool

**Default value**

0x0

**Description**

Enable exception on the Main/Epilogue instruction when executed after a mismatched Prologue/Main in a CPY/SET sequence, or after another random instruction.

**mops\_inst\_cpy\_zero\_size\_can\_fault****Type**

bool

**Default value**

0x1

**Description**

Fault because of mismatched implementation option when inst\_cpy\_size is 0.

**mops\_inst\_set\_zero\_size\_can\_fault****Type**

bool

**Default value**

0x1

**Description**

Fault because of mismatched implementation option when inst\_set\_size is 0.

**mops\_mismatched\_page\_crossing\_access\_unpred****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable behaviour for FEAT\_MOPS instructions when crossing page boundary with different memory types, 0 : Memory block access uses the attributes of it's own address block 1: Alignment Fault.

**mops\_mmu\_abort\_far\_aligned****Type**

bool

**Default value**

0x0

**Description**

If true, in case of an MMU abort on a MOPS instruction, report FAR aligned to current translation granule.

**mops\_set\_block\_size****Type**

int

**Default value**

0x40

**Description**

Block size used for memset memory accesses.

**mops\_set\_handle\_async\_exceptions****Type**

bool

**Default value**

0x0

**Description**

Handle any pending async exceptions after setting a block of data, instead of waiting until instruction end.

**mops\_set\_post\_size****Type**

int

**Default value**

0xa

**Description**

Percentage of data copied in memset 'E' instructions.

**mops\_set\_pre\_size****Type**

int

**Default value**

0xa

**Description**

Percentage of data copied in memset 'P' instructions.

**mops\_set\_single\_access****Type**

bool

**Default value**

0x0

**Description**

Execute memset as a single read and single write access.

**mops\_set\_zero\_size\_can\_fault****Type**

bool

**Default value**

0x1

**Description**

Fault because of mismatched implementation option when the operation is of size 0.

**mops\_setg\_unaligned\_does\_mismatch\_fault****Type**

bool

**Default value**

0x0

**Description**

If true, in case of unaligned SETGM / SETGE, raise a mismatched memset exception because of impdef reasons, instead of alignment fault.

**mops\_wp\_far\_behaviour****Type**

int

**Default value**

0x0

**Description**

Set option for address stored in FAR/EDWARD after watchpoints hit by MOPS instructions  
0 - FAR recorded matches lowest watchpointed address accessed by the instruction  
1 - FAR recorded matches lowest address accessed by the instruction within same translation granule as watchpointed address  
2 - FAR recorded matches highest watchpointed address accessed by the instruction that triggered the watchpoint.

**mpam\_bw\_bwa\_wd****Type**

int

**Default value**

0x1

**Description**

MPAM MPAMBWIDR\_EL1.BWA\_WD: The number of implemented bits in the bandwidth allocation fields {MPAMBWn\_ELx, MPAMBWSM\_EL1}.MAX and MPAMBWCAP\_EL2.CAP.

**mpam\_bw\_has\_hw\_scale****Type**

bool

**Default value**

0x0

**Description**

MPAM Whether has hardware support for auto-scaling of {MPAMBWn\_ELx, MPAMBWSM\_EL1}.MAX and MPAMBWCAP\_EL2.CAP limits.

**mpam\_bw\_max\_lim****Type**

int

**Default value**

0x0

**Description**

MPAM the implemented maximum-bandwidth limit partitioning behaviors: - 0, Both soft limit and hard limit behaviors are implemented. - 1, Soft limit behavior is implemented. - 2, Hard limit behavior is implemented.

**mpam\_bw\_us\_frac****Type**

int

**Default value**

0x0

**Description**

MPAM MPAMBWIDR\_EL1.US\_FRAC: The fractional part of the window width in microseconds.

**mpam\_bw\_us\_int****Type**

int

**Default value**

0x0

**Description**

MPAM MPAMBWIDR\_EL1.US\_INT: The integer part of the window width in microseconds.

**mpam\_force\_ns\_rao****Type**

bool

**Default value**

0x0

**Description**

Whether MPAM3\_EL3.FORCE\_NS bit is RAO/WI.

**mpam\_frac****Type**

int

**Default value**

0x0

**Description**

MPAM fractional revision number in ID\_AA64PFR1\_EL1.MPAM\_frac field. Combines with has\_mpam to give the mpam version has\_mpam = false, mpam\_frac = 0 -> Not implemented  
has\_mpam = false, mpam\_frac = 1 -> FEAT\_MPAMvOp1 has\_mpam = true, mpam\_frac = 0 -> FEAT\_MPAMv1p0 has\_mpam = true, mpam\_frac = 1 -> FEAT\_MPAMv1p1.

**mpam\_has\_altsp****Type**

bool

**Default value**

0x0

**Description**

MPAM Whether MPAMIDR\_EL1.HAS\_ALTSP bit is set or clear.

**mpam\_has\_bw\_ctrl****Type**

bool

**Default value**

0x0

**Description**

MPAM Whether MPAMIDR\_EL1.HAS\_BW\_CTRL bit is set or clear.

**mpam\_has\_hcr****Type**

bool

**Default value**

0x0

**Description**

MPAM Whether MPAMIDR\_EL1 HAS\_HCR bit is set or clear.

**mpam\_max\_partid****Type**

int

**Default value**

0xffff

**Description**

MPAM Maximum PARTID Supported.

**mpam\_max\_pmg**

**Type**

int

**Default value**

0xff

**Description**

MPAM Maximum PMG Supported.

**mpam\_max\_vpmr**

**Type**

int

**Default value**

0x0

**Description**

MPAM Maximum VPMR Supported.

**mpamidr\_has\_force\_ns**

**Type**

int

**Default value**

0x0

**Description**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpamidr\_has\_sdeflt**

**Type**

int

**Default value**

0x0

**Description**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.



**mpamidr\_has\_tidr****Type**

int

**Default value**

0x0

**Description**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear Possible values of this parameter are:

- 0, feature is not enabled.
- 1, feature is implemented if ARMv8.6 is enabled.
- 2, feature is implemented.

**mpidr\_layout****Type**

int

**Default value**

0x0

**Description**

Layout of MPIDR. 0 AFF0 is CPUID, 1 AFF1 is CPUID.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**mpmm\_config****Type**

string

**Default value**

""

**Description**

MPMMTUNE register value. The JSON schema is : { "DT\_THR":0, "MPMM\_G2\_TP":2, "MPMM\_G2\_ATHR":100, "MPMM\_G1\_TP":2, "MPMM\_G1\_ATHR":75, "MPMM\_GO\_TP":2, "MPMM\_GO\_ATHR":50 } . The value given for threshold value is just an indication, not specific to any core. This parameter is used only when has\_mpmm is set.

**mte\_report\_which\_failed\_address****Type**

string

**Default value**

"first, mops\_first\_failing\_address\_in\_range"

**Description**

Set to <OPT>, <MOPS\_OPT>Applicable only for MTE synchronous check. OPT defines the range for the failing address to report and MOPS\_OPT defines the choice within that range for MOPS operations only. Non-MOPS operations report the first address in the range defined by OPT. OPT is set to "first" or "last". If "first" then report an address from the intersection of the first failed MTE granule and the transaction's range. If "last" then report an address from the intersection of the last failed MTE granule and the transaction's range. MOPS\_OPT is set to "mops\_first\_failing\_address\_in\_range" or "mops\_random\_address\_in\_range". If "mops\_first\_failing\_address\_in\_range" then report the first failing address in the range defined by OPT. If "mops\_random\_address\_in\_range" then report a random address within the range defined by OPT.

**mte\_tminline****Type**

int

**Default value**

0x0

**Description**

Value of CTR\_ELO.TminLine for reading purpose only. A value configured using this does not indicate the presence of separate tag cache. 0, TminLine evaluated from smallest data cache line.

**mvbar\_reset\_is\_rvbar****Type**

bool

**Default value**

0x1

**Description**

If true then the reset value of MVBAR is RVBAR, if false the reset value is UNKNOWN.

**non\_secure\_vgic\_alias\_when\_ns\_only****Type**

int

**Default value**

0x0

**Description**

If ! has\_el3 and only non-secure side exists, then the normal position of the VGIC is a secure alias. If this parameter is non-zero then in addition a non-secure alias of the VGIC will be placed at this position (aligned to 32 KB).

**num\_loregion\_descriptors****Type**

int

**Default value**

0x0

**Description**

Number of Limited Ordering Region descriptors implemented (if ARM v8.1 extensions are implemented) (FEAT\_LOR).

**num\_loregions****Type**

int

**Default value**

0x0

**Description**

Number of Limited Ordering Regions implemented excluding background region (if ARM v8.1 extensions are implemented) (FEAT\_LOR).

**number\_of\_abl\_breakpoints****Type**

int

**Default value**

0x0

**Description**

if FEAT\_ABLE is implemented, Number of address matching breakpoints that support address linking.

**number\_of\_error\_records****Type**

int

**Default value**

0x0

**Description**

Cores Number of Error records supported for RAS.

**nv\_frac\_support\_level****Type**

int

**Default value**

0x0

**Description**

Support for a subset of FEAT\_NV and FEAT\_NV2 behaviours: 0 - Not implemented. 1 - Implemented. 2 - Implemented with FEAT\_NV2p1.

**optimal\_alignment\_size****Type**

int

**Default value**

0x1

**Description**

Alignment boundary which does not incur additional performance penalty from ARMv8.5. - 1, architectural misalignment is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11] - 2, access crossing 4 byte boundary is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11] - 3, access crossing 8 byte boundary is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11] ... - 12, access crossing 4 Kbyte boundary is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11] .

**output\_attributes****Type**

string

**Default value**

"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID,  
ExtendedID[38]=MPAM\_NS"

**Description**

User-defined transform to be applied to bus attributes like MasterID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

**pacm\_support\_level****Type**

int

**Default value**

0x2

**Description**

Implements PSTATE.PACM from ARMv9.5. 0: Not supported, 1: Trivial implementation when FEAT\_PAuth\_LR and FEAT\_PACIMP are supported, 2: Full implementation when FEAT\_PAuth\_LR is supported.

**page\_based\_hardware\_attributes****Type**

int

**Default value**

0x0

**Description**

Implement the page based hardware attributes from ARMv8.2. This parameter indicates which page table bits are available for hardware, where bits[3:0] correspond to PTE[62:59] and to TCR\_ELx.HWUnyy (FEAT\_HPDS2).

**pan\_stage1\_in\_realm\_el2\_0\_is\_uxn****Type**

bool

**Default value**

0x0

**Description**

If FEAT\_PAN3 is implemented, whether stage1 translation in the Realm EL2&0 regime that resolves to a NS address is treated as unprivileged execute-never.

**par\_ns\_set\_unknown\_bit****Type**

bool

**Default value**

0x1

**Description**

Whether NS bit of PAR is set/clear when executing AT to perform non-secure regime translation. When true, NS is set to 1 else 0.

**par\_nse\_set\_unknown\_bit****Type**

bool

**Default value**

0x0

**Description**

Whether NSE bit of PAR is set/clear when executing AT operation on secure, non-secure or realm translation regime. When true, NSE is set to 1 else 0.

**pfar\_is\_valid****Type**

bool

**Default value**

0x1

**Description**

IMPLEMENTATION DEFINED choice to configure ESR\_ELx.PFV: whether PFAR\_ELx is valid or UNKNOWN when ESR\_ELx.PFV is not forced to be 0.

**pfr1\_csv2\_frac****Type**

int

**Default value**

0x0

**Description**

Fractional revision number ID\_AA64PFR1\_EL1.CSV2\_frac when ID\_AA64PFR0\_EL1.CSV==1 for CSV2 extension (FEAT\_CSV2\_1p1, FEAT\_CSV2\_1p2).

**pmb\_idr\_external\_abort****Type**

int

**Default value**

0x0

**Description**

Describes how the PE manages External aborts on writes made by the Statistical Profiling Extension to the Profiling Buffer. 0, External abort is reported to SPE, From Armv8.8 and Armv9.3, the value 0 is not permitted. 1, External abort is ignored. 2, The External abort generates an SError and the error is not reported to SPE.

**pmb\_idr\_flag\_updates****Type**

bool

**Default value**

0x1

**Description**

Defines whether the address translation performed by the Profiling Buffer manages the Access Flag and dirty state.

**pmbstr\_reports\_external\_abort****Type**

bool

**Default value**

0x1

**Description**

Whether PMBSR\_ELx.EA and PMBSR\_ELx.DL are set as the result of an external abort or are treated as RES0.

**pmcr\_disable\_events\_export****Type**

bool

**Default value**

0x1

**Description**

If true, export for PMU events is disabled. This configures PMCFGR.EX field.

**pmmir\_el1\_bus\_slots****Type**

int

**Default value**

0x0

**Description**

Largest value by which BUS\_ACCESS can increment over BUS\_CYCLES cycles. From v8.7 PMU extension.

**pmmir\_el1\_bus\_width****Type**

int

**Default value**

0x0

**Description**

Width, in bytes, of accesses counted by BUS\_ACCESS. From v8.7 PMU extension.

**pms\_idr\_max\_size****Type**

int

**Default value**

0x6

**Description**

Defines largest size for a single SPE record (rounded up to a power of 2).

**pmu-num\_counters****Type**

int

**Default value**

0x8

**Description**

Number of PMU counters implemented.

**`pmu_async_exception_delay`****Type**

int

**Default value**

0x0

**Description**

Configure PMU asynchronous exception delay in CPU cycles (FEAT\_SEBEP).

**`pmu_cycle_counter_counts_actual_cycles`****Type**

bool

**Default value**

0x0

**Description**

If true and Timing annotation is enabled, PMU cycle counter counts actual cycles, otherwise counts instructions executed.

**`pmu_has_chain_event`****Type**

bool

**Default value**

0x1

**Description**

PMU (if present) implements event number 0x1e, CHAIN.

**`pmu_precise_events`****Type**

string

**Default value**

""

**Description**

"Configure v9.4 Precise PMU events. {"pmu\_events":["SW\_INCR", "PC\_WRITE\_RETIRED", "BR\_RETIRED", "BR\_IND\_RETIRED", "BR\_RETURN\_RETIRED", "BR\_RETURN\_ANY\_RETIRED", "BR\_IND\_TAKEN\_RETIRED", "LD\_RETIRED", "ST\_RETIRED", "UNALIGNED\_LD\_ST", "INST\_RETIRED", "EXCEP\_TAKEN", "EXCEP\_RETURN", "CHAIN"]}".



**pmu\_threshold\_bit\_width****Type**

int

**Default value**

0x0

**Description**

Implement FEAT\_PMUv3\_TH and if so the width of PMEVTPER<n>\_ELO.TH in bits. 0, not implemented. 1-12 number of bits in PMEVTPER<n>\_ELO.TH.

**poison\_range\_end\_addr****Type**

int

**Default value**

0x0

**Description**

End PA of poisoned range.

**poison\_range\_start\_addr****Type**

int

**Default value**

0x0

**Description**

Start PA of poisoned range.

**pseudo\_fault\_generation\_feature\_register****Type**

string

**Default value**

""

**Description**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_pseudo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

**pstate\_pm\_reset****Type**

bool

**Default value**

0x0

**Description**

Reset value of PSTATE.PM.

**pstate\_ssbs\_reset****Type**

bool

**Default value**

0x0

**Description**

Reset value of pstate.ssbs.

**pstate\_ssbs\_type****Type**

int

**Default value**

0x0

**Description**

Implement speculative store bypass safe feature from ARMv8.5. 0, Not supported. 1, Supported without MSR/MRS access to SSBS (FEAT\_SSBS). 2, fully supported (FEAT\_SSBS2).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**randomize\_unknowns\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Will fill in unknown bits in registers at reset with random value using register\_reset\_data as seed, it overrides scramble\_unknowns\_at\_reset.

**ras\_aderr\_anerr\_controls\_are\_same****Type**

bool

**Default value**

0x0

**Description**

If true and FEAT\_ADERR and FEAT\_ANERR is implemented then ADERR and ANERR controls should always be set to the same value (FEAT\_ADERR) (FEAT\_ANERR).

**ras\_aderr\_applies\_to\_nv2****Type**

bool

**Default value**

0x1

**Description**

If true and FEAT\_ADERR is implemented, NV2-transformed accesses to device memory may generate synchronous errors. If false, they only generate asynchronous errors (FEAT\_ADERR) (FEAT\_NV2).

**ras\_anerr\_applies\_to\_nv2****Type**

bool

**Default value**

0x1

**Description**

If true and FEAT\_ANERR is implemented, NV2-transformed accesses to normal memory may generate synchronous errors. If false, they only generate asynchronous errors (FEAT\_ANERR) (FEAT\_NV2).

**ras\_err\_registers\_undef\_if\_no\_error\_records****Type**

bool

**Default value**

0x0

**Description**

If true, all RAS error record registers, along with ERRSELR\_EL1, will be undefined if ERRIDR\_EL1 indicates that zero error records are implemented.

**ras\_errselr\_undef\_if\_no\_error\_records****Type**

bool

**Default value**

0x0

**Description**

If true, ERRSELR\_EL1 will be undefined if ERRIDR\_EL1 indicates that zero error records are implemented.

**ras\_extra\_configurations****Type**

string

**Default value**

""

**Description**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCN masks - these are 64 bit masks covering the 64 bit registers ERXMISCN\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

**ras\_frac****Type**

int

**Default value**

0x0

**Description**

0, No additional feature implemented. 1, Additional ERXMISC\*, ERXPFG\* registers and FaultInjection trap from RAS v1.1. implemented.

**ras\_log2\_fault\_granule\_size****Type**

int

**Default value**

0xc

**Description**

Log2 of the RAS fault granule size in Bytes.

**ras\_mmap\_address****Type**

string

**Default value**

""

**Description**

Base address of memory mapped RAS Registers for each core on system bus. 0 means the RAS is not mapped, otherwise the address must be at least 4KB aligned or more depending upon the features implemented. JSON schema for the parameter value is: {"format": "all\_addrs\_are\_absolute\_wrt\_systembus", "cores": [{"ras": 0x0}, {"ras": 0x0}, {"ras": 0x0}, {"ras": 0x0}]}.

**ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0x18

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**ras\_report\_aligned\_pa\_in\_pfar****Type**

bool

**Default value**

0x0

**Description**

If true, the PFAR\_ELx register reports the PA aligned to the RAS fault granule size on a sync external abort or SError exception.

**register\_reset\_data****Type**

int

**Default value**

0x0

**Description**

Data used to fill register bits when they become UNKNOWN at reset.

**register\_reset\_data\_hi****Type**

int

**Default value**

0x0

**Description**

Data used to fill the upper-half of 128-bit registers when the bits become UNKNOWN at reset.

**report\_iside\_cmo\_ifsr****Type**

bool

**Default value**

0x1

**Description**

fault info for an iside cache maintenance operation is reported in the IFSR.

**report\_second\_access\_align\_fault\_non\_atomic\_pair\_access****Type**

bool

**Default value**

0x0

**Description**

If true, the non-atomic load/store pair accesses report the 2nd register as faulting address for an alignment fault. This is IMP-DEF behavior as defined in FEAT\_LRCPC3.

**report\_second\_access\_mmu\_fault\_non\_atomic\_pair\_access****Type**

bool

**Default value**

0x0

**Description**

If true, the non-atomic load/store pair accesses report the 2nd register as faulting address for an MMU fault. This is IMP-DEF behavior as defined in FEAT\_LRCPC3.

**reported\_fp\_revision****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reserved\_HMC\_SSC\_PAC\_treated\_disabled****Type**

bool

**Default value**

0x0

**Description**

When DBG[B|W]CR.{HMC,SSC,PAC} bits configuration is reserved, this parameter controls whether breakpoints/watchpoints are treated as Disabled or not.

**restore\_fpsr\_on\_trapped\_fp\_exception****Type**

bool

**Default value**

0x0

**Description**

If true, FPSR is restored to the value of the FPSR immediately before the instruction that generated the trapped floating-point exception.

**restriction\_on\_speculative\_execution****Type**

int

**Default value**

0x0

**Description**

Implements the ARMv8.5 security feature (Restrictions on the effects of speculation), ID\_AA64PFR0\_EL1.CSV2: 0: No disclosure whether branch targets trained in one context can affect speculative execution in a different context, 1: Branch targets trained in one context cannot affect speculative execution in a different hardware described context (SCXTNUM\_ELx not supported), 2: Branch targets trained in one context cannot affect speculative execution in a different hardware described context (SCXTNUM\_ELx supported) (FEAT\_CSV2, FEAT\_CSV2\_2), 3: FEAT\_CSV2\_3 is supported.

**restriction\_on\_speculative\_execution\_aarch32****Type**

int

**Default value**

0x0

**Description**

Implements the ARMv8.5 security feature (Restrictions on the effects of speculation), ID\_PFR0.CSV2: 0: No disclosure whether branch targets trained in one context can affect speculative execution in a different context, 1: Branch targets trained in one context cannot affect speculative execution in a different hardware described context, 2: Branch targets trained in one context cannot affect speculative execution in a different hardware described context or at a different address in the same hardware described context (FEAT\_CSV2, FEAT\_CSV2\_2).

**revision\_number****Type**

int

**Default value**

0x0

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.



**rgsr\_res0\_stateful****Type**

bool

**Default value**

0x0

**Description**

Whether RGSr read of RES0 bits return last written value.

**rme\_default\_mecid\_nonsecure****Type**

int

**Default value**

0x0

**Description**

Default MECID value for NON-SECURE PAS.

**rme\_default\_mecid\_realm****Type**

int

**Default value**

0x0

**Description**

Default MECID value for REALM PAS.

**rme\_default\_mecid\_root****Type**

int

**Default value**

0x0

**Description**

Default MECID value for ROOT PAS.

**rme\_default\_mecid\_secure****Type**

int

**Default value**

0x0

**Description**

Default MECID value for SECURE PAS.

**rme\_full\_is\_tagged\_nsh\_treated\_as\_tagged****Type**

bool

**Default value**

0x0

**Description**

Whether a tagged NonShared memory attribute is treated as tagged or not. Does nothing if effective RME support is not full.

**rme\_level0\_gpt\_size****Type**

int

**Default value**

0x0

**Description**

The range of address space protected by each entry in the level 0 GPT (0->1GB 1->16GB, 2->64GB, 3->512GB).

**rme\_mecid\_width****Type**

int

**Default value**

0x1

**Description**

Width of MECID in bits.

**rme\_nsh\_cacheable\_is\_shareable****Type**

bool

**Default value**

0x0

**Description**

If true, NSH cacheable becomes shareable cacheable (FEAT\_RME).

**rme\_support\_level****Type**

int

**Default value**

0x0

**Description**

0 -> Realm management extension not implemented, 1 -> LEGACY\_TZ\_EN mode i.e. RME register fields are stateful but only supports secure/non-secure states, 2 -> Realm management extension fully implemented (FEAT\_RME).

**`rnr_always_implemented`****Type**

bool

**Default value**

0x0

**Description**

Always implement RMR\_ELx, RMR, or HRMR at the highest implemented exception level, even if that exception level cannot use both AArch32 and AArch64.

**`rndr_rndrrs_seed`****Type**

int

**Default value**

0x0

**Description**

Initial seed for random engine used in RNDR register.

**`s1_align_memtype_fault_prio_more_than_s2_perm_fault_on_s1_walk`****Type**

bool

**Default value**

0x1

**Description**

If true, s1 alignment fault has priority over s2 permission faults.

**`s1_perm_fault_prio_more_than_s2_perm_fault_on_s1_walk`****Type**

bool

**Default value**

0x0

**Description**

If true, s1 permission fault has priority over s2 on s1 translation table walk permission faults.

**scheduler\_mode****Type**

int

**Default value**

0x0

**Description**

Control the interleaving of instructions in this processor. 0, default long quantum. 1, low latency mode, short quantum and signal checking. 2, lock-breaking mode, long quantum with additional context switches near load-exclusive instructions. WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

**scr\_nET\_writeable****Type**

bool

**Default value**

0x0

**Description**

Whether SCR.nET is writeable. Writing to it is purely cosmetic (nET behavior not implemented).

**scramble\_unknowns\_at\_reset****Type**

bool

**Default value**

0x1

**Description**

Will fill in unknown bits in registers at reset with register\_reset\_data.

**serror\_clear\_delay****Type**

int

**Default value**

0x0

**Description**

Delay for clearing of SError if SError is level-triggered, in cpu cycles.

**set\_mops\_option****Type**

int

**Default value**

0x0

**Description**

Set option for Armv8.8 SET(FEAT\_MOPS). 0, use default(i.e. use value configured through has\_mops\_option). 1, implemented using Option A. 2, implemented using Option B.

**set\_rasv10\_for\_armv84\_and\_higher****Type**

bool

**Default value**

0x0

**Description**

ARMv8.4 mandates RAS System Architecture v1.1, but when there are no error records and FEAT\_DoubleFault is not implemented then there is no functional difference between the RAS System Architecture v1.0 (that is, the RAS extension as in pre-ARMv8.4 implementations) and the RAS System Architecture v1.1 (also known as FEAT\_RASv1p1). This flag if true will set the RAS ID to declare RAS v1.0 rather than RAS v1.1 for ARMv8.4 and higher implementations. If this is set and the core does not conform to the restrictions then this parameter is ignored.

**setg\_mops\_option****Type**

int

**Default value**

0x0

**Description**

Set option for Armv8.8 SETG(FEAT\_MOPS). 0, use default(i.e. use value configured through has\_mops\_option). 1, implemented using Option A. 2, implemented using Option B.

**skip\_trace\_on\_write\_to\_oseccr\_el1\_when\_oslock\_is\_unlocked****Type**

bool

**Default value**

0x0

**Description**

If OSLSR\_EL1.OSLK == 0, then OSECCR\_EL1 returns an unknown value on reads and ignores writes. When true, also skips the traces on writes to OSECCR\_EL1 when OSLSR\_EL1.OSLK == 0.

**spe\_counter\_size****Type**

int

**Default value**

0x1

**Description**

Size of counter packet payload in Statistical Profiling Extension - 1, Counter packet payloads are 12-bit saturating counters - 2, Counter packet payloads are 16-bit saturating counters.

**spmu\_support\_level****Type**

int

**Default value**

0x0

**Description**

Implement System PMU: 0: Not supported, 1: v8.9 System PMU Extension is implemented (FEAT\_SPMU), 2: v9.5 System PMU2 Extension is implemented (FEAT\_SPMU2) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.9 is enabled. - 2, feature is implemented.

**spsr\_el3\_is\_mapped\_to\_spsr\_mon****Type**

bool

**Default value**

0x1

**Description**

Whether SPSR\_EL3 is mapped to AArch32 register SPSR\_mon.

**spsr\_m4\_res0****Type**

bool

**Default value**

0x0

**Description**

Whether SPSR\_ELx.M[4] bit should be RES0 for AARCH64 only implementations.

**stage12\_tlb\_size****Type**

int

**Default value**

0x400

**Description**

Number of stage1+2 tlb entries.

**stage1\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1 only tlb entries.

**stage1\_walkcache\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1 only walk cache entries.

**stage2\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage2 only tlb entries.

**stage2\_walkcache\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage2 only walk cache entries.

**statistical\_profiling\_buffer\_alignment****Type**

int

**Default value**

0x1

**Description**

Statistical profiling alignment constraint for sample buffer.

**statistical\_profiling\_random\_interval\_is\_separate****Type**

bool

**Default value**

0x0

**Description**

Statistical profiling random interval gets added to the main timer interval(false) or (true) runs as separate timer.

**statistical\_profiling\_recommended\_min\_sampling****Type**

int

**Default value**

0x100

**Description**

Statistical profiling recommended minimum sampling interval.

**stex\_fail\_suppress\_sync\_data\_aborts****Type**

bool

**Default value**

0x0

**Description**

If true, synchronous data aborts are not reported if store exclusive fails.

**store\_excl\_fail\_tag\_check\_action****Type**

int

**Default value**

0x0

**Description**

Behavior of tag check by core when a store exclusive fails. 0, Tag check ignored, 1, Tag check done if exclusive fails by global monitor.

**strex\_fail\_can\_hit\_watchpoint****Type**

bool

**Default value**

0x0



**Description**

If true, a strex fail can hit watchpoint.

**stzgm\_reports\_fault\_address\_from\_reg\_arg****Type**

int

**Default value**

0x0

**Description**

Which faulting address should be reported in FAR\_ELx on a failed STZGM: 0: the lowest aligned addr to DCZID-log2-block-size, 1: the addr held in the register argument, 2: if it is a tag-check fault, the addr aligned to DCZID-log2-block-size, otherwise the addr held in the register argument.

**supports\_multi\_threading****Type**

bool

**Default value**

0x0

**Description**

Sets the MPIDR.MT bit. Setting this to true hints the the cluster is multi-threading compatible.

**sve.clear\_constrained\_lanes****Type**

int

**Default value**

0x0

**Description**

When a constrained vector length increases, previously inaccessible bits are set to zero. Possible values are: 0=never, 1=always, 2=if the register was written to while the vector length was constrained.

**sve.combine\_movprfx\_and\_destructive****Type**

bool

**Default value**

0x0

**Description**

Attempt to combine the execution of MOVPRFX and the destructively-encoded instruction that follows it.

**sve.disable\_speculative\_accesses****Type**

bool

**Default value**

0x0

**Description**

All speculative memory accesses behave as though faulting, without accessing memory.

**sve.enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Start with system registers set up for Scalable Vector Extension use.

**sve.ffr\_16b\_pattern\_UNKNOWN****Type**

int

**Default value**

0x0

**Description**

A specific 16-bit UNKNOWN value that is used by parameter force\_UNKNOWN\_to\_ffr.

**sve.force\_UNKNOWN\_to\_ffr****Type**

int

**Default value**

0x0

**Description**

Governs behavior if WRFFR writes a non-monotonic value to FFR. Possible values are: 0 - Write non-canonical value to FFR, 1 - Overwrite FFR with a specific pattern of 16-bit UNKNOWN value. See ffr\_16b\_pattern\_UNKNOWN, 2 - Clear all bits above first zero 3 - Set all bits after first one.

**sve.fp\_exception\_report\_lowest****Type**

bool

**Default value**

0x0

**Description**

If true, for multiple trapped FP exceptions, report the lowest lane in VECITR. Otherwise, report the highest.

**sve.fp\_exception\_set\_tfv****Type**

bool

**Default value**

0x1

**Description**

Set ESR\_ELx.TFV during FP exception. Trapped exception flags are valid.

**sve.fp\_exception\_set\_vecitr****Type**

bool

**Default value**

0x0

**Description**

If true, set ESR\_ELx.VECITR during FP exception. Otherwise, set RES0.

**sve.has\_b16b16****Type**

int

**Default value**

0x0

**Description**

Whether FEAT\_SVE\_B16B16 is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if either FEAT\_SVE2 or FEAT\_SME2 is implemented.

**sve.has\_bfscale****Type**

int

**Default value**

0x0

**Description**

Whether FEAT\_SVE\_BFSCALE is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if either FEAT\_SVE2 or FEAT\_SME2 is implemented.

**sve.has\_sme****Type**

bool

**Default value**

0x0

**Description**

Whether SME is implemented (FEAT\_SME).

**sve.has\_sme2****Type**

bool

**Default value**

0x0

**Description**

Whether SME2 is implemented (FEAT\_SME2).

**sve.has\_sme\_b16b16****Type**

int

**Default value**

0x0

**Description**

Whether FEAT\_SME\_B16B16 is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if FEAT\_SME2 is implemented.

**sve.has\_sme\_f16f16****Type**

int

**Default value**

0x0

**Description**

Whether FEAT\_SME\_F16F16 is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if FEAT\_SME2 is implemented.

**sve.has\_sme\_f64f64****Type**

int

**Default value**

0x1

**Description**

If SME is implemented, whether double-precision FMOPA and FMOPS are implemented.

**sve.has\_sme\_f8f16**

**Type**

int

**Default value**

0x1

**Description**

If SME2 is implemented, whether FEAT\_SME\_F8F16 is implemented.

**sve.has\_sme\_f8f32**

**Type**

int

**Default value**

0x1

**Description**

If SME2 is implemented, whether FEAT\_SME\_F8F32 is implemented.

**sve.has\_sme\_fa64**

**Type**

bool

**Default value**

0x0

**Description**

Whether FEAT\_SME\_FA64 is implemented.

**sve.has\_sme\_i16i64**

**Type**

int

**Default value**

0x1

**Description**

If SME is implemented, whether instructions that accumulate 16-bit integer outer products into 64-bit integer tiles are implemented.

**sve.has\_sme\_lutv2**

**Type**

bool

**Default value**

0x0

**Description**

Whether FEAT\_SME\_LUTv2 is implemented.

**sve.has\_sme\_mop4****Type**

int

**Default value**

0x0

**Description**

Whether FEAT\_SME\_MOP4 is implemented. Possible values are 0 - Implemented if FEAT\_SME2p2 is implemented, 1 - Implemented if FEAT\_SME2p1 is implemented.

**sve.has\_sme\_priority\_control****Type**

bool

**Default value**

0x1

**Description**

Whether SME Priority Control is implemented.

**sve.has\_sme\_tmop****Type**

int

**Default value**

0x0

**Description**

Whether FEAT\_SME\_TMOP is implemented. Possible values are 0 - Implemented if FEAT\_SME2p2 is implemented, 1 - Implemented if FEAT\_SME2p1 is implemented.

**sve.has\_ssve\_aes****Type**

int

**Default value**

0x0

**Description**

Indicates support for SVE2 and SME2 AES instructions when the PE is in Streaming SVE mode (FEAT\_SSVE\_AES).

**sve.has\_ssve\_bit\_perm****Type**

int

**Default value**

0x0

**Description**

Whether FEAT\_SSVE\_BitPerm is implemented. Possible values are 0 - Implemented if FEAT\_SME2p2 is implemented, 1 - Implemented if FEAT\_SME2p1 is implemented.

**sve.has\_sve2****Type**

bool

**Default value**

0x0

**Description**

Whether SVE2 is implemented (FEAT\_SVE2).

**sve.has\_sve2\_aes****Type**

int

**Default value**

0x2

**Description**

If SVE2 is implemented, whether AES instructions are implemented. Possible values are: 0 - not implemented, 1 - SVE2 AESE, AESD, AESMC, and AESIMC are implemented (FEAT\_SVE\_AES), 2 - Same as 1 but in addition SVE2 PMULLB and PMULLT with 64-bit source are implemented, 3 - Same as 2 but SVE2 64-bit source element PMLALB and PMLALT instruction variants are implemented (FEAT\_SVE\_PMULL128).

**sve.has\_sve2\_bit\_perm****Type**

bool

**Default value**

0x1

**Description**

If SVE2 is implemented, whether BitPerm instructions are implemented (FEAT\_SVE\_BitPerm).

**sve.has\_sve2\_sha3****Type**

bool

**Default value**

0x1

**Description**

If SVE2 is implemented, whether SHA3 instructions are implemented (FEAT\_SVE\_SHA3).

**sve.has\_sve2\_sm4****Type**

bool

**Default value**

0x1

**Description**

If SVE2 is implemented, whether SM4 instructions are implemented (FEAT\_SVE\_SM4).

**sve.has\_sve\_bf16****Type**

bool

**Default value**

0x1

**Description**

Whether SVE BFloat16 instructions are implemented.

**sve.has\_sve\_extended\_bf16****Type**

int

**Default value**

0x2

**Description**

Deprecated: to enable FEAT\_EBF16, use CPU parameter has\_ebf16. Whether Extended BFloat16 instructions are implemented. Possible values are: 0 - Disabled, 1 - Enabled if SME or SVE is implemented, 2 - Enabled if SME is implemented.

**sve.has\_sve\_f16f32mm****Type**

bool

**Default value**

0x0

**Description**

Whether the SVE half-precision to single-precision Matrix Multiply instructions are implemented (FEAT\_F16F32MM).



**sve.has\_sve\_mm\_f32****Type**

bool

**Default value**

0x1

**Description**

Whether the SVE FP32 Matrix Multiply instructions are implemented (FEAT\_F32MM).

**sve.has\_sve\_mm\_f64****Type**

bool

**Default value**

0x1

**Description**

Whether the SVE FP64 Matrix Multiply instructions are implemented (FEAT\_F64MM).

**sve.has\_sve\_mm\_i8****Type**

bool

**Default value**

0x1

**Description**

Whether the SVE Int8 Matrix Multiply instructions are implemented (FEAT\_I8MM).

**sve.movprfx\_unpredictable\_behavior****Type**

int

**Default value**

0x0

**Description**

Defines the behavior of MOVPRFX and the instruction it immediately precedes when the behavior is CONSTRAINED UNPREDICTABLE. Possible values are: 0 - UNDEF execution from MOVPRFX, 1 - MOVPRFX and second half of instruction executes as NOP, 2 - NOP MOVPRFX only, 3 - UNDEF execution from MOVPRFX unless otherwise trapped.

**sve.predicated\_sp\_align\_check\_behaviour****Type**

int

**Default value**

0x0

**Description**

Governs behavior of SP alignment checking for predicated memory accesses. Possible values are: 0 - Always perform, 1 - Skip if governing predicate is 0, 2 - Skip for contiguous accesses if governing predicate is 0, 3 - Skip for gather/scatter accesses if governing predicate is 0.

**sve.relax\_sme\_watchpoint\_matching\_16****Type**

bool

**Default value**

0x0

**Description**

Whether memory accesses through Z and P registers in Streaming Mode and all accesses through ZA match watchpoints rounded to 16-byte alignment.

**sve.relax\_sve\_watchpoint\_matching\_16****Type**

bool

**Default value**

0x0

**Description**

If FEAT\_DEBUGv8p9 is implemented, whether memory accesses through Z and P registers outside Streaming Mode match watchpoints rounded to 16-byte alignment.

**sve.sm\_tag\_checked****Type**

bool

**Default value**

0x1

**Description**

Whether SME, SVE, and SIMD&FP load and store instructions executed when the PE is in Streaming SVE mode perform a Tag Check.

**sve.sme2\_version****Type**

int

**Default value**

0x0

**Description**

The version of SME2 if implemented. Possible values are: 0 - FEAT\_SME2, 1 - FEAT\_SME2p1, 2 - FEAT\_SME2p2.

**sve.sme\_highest\_implemented\_priority****Type**

int

**Default value**

0x0

**Description**

When SME Priority Control and SME2p2 are implemented, controls the highest implemented priority.

**sve.sme\_only****Type**

bool

**Default value**

0x0

**Description**

If SME is implemented, whether SVE functionality is available only when SM=1.

**sve.sme\_ssve\_fp8\_support\_level****Type**

int

**Default value**

0x0

**Description**

If FEAT\_SME2 and FEAT\_FP8 are implemented, whether FP8 operations are supported in Streaming Mode where not implemented outside Streaming Mode. Possible values are: 0 - No support above FEAT\_FP8, 1 - FEAT\_SSVE\_FP8FMA, 2 - FEAT\_SSVE\_FP8DOT4, 3 - FEAT\_SSVE\_FP8DOT2.

**sve.sme\_veclens\_implemented****Type**

int

**Default value**

0x7

**Description**

Which SME vector lengths are implemented. Represented as a bitfield where bit[n]==1 implies SME vector length of  $128 \cdot 2^n$  bits is implemented.

**sve.smidr\_el1\_implementer\_val****Type**

int

**Default value**

0x41

**Description**

The value of SMIDR\_EL1.Implementer.

**sve.smidr\_el1\_nsmc\_val****Type**

int

**Default value**

0x0

**Description**

The value of SMIDR\_EL1.NSMC.

**sve.smidr\_el1\_revision\_val****Type**

int

**Default value**

0x0

**Description**

The value of SMIDR\_EL1.Revision.

**sve.smidr\_el1\_sh\_val****Type**

int

**Default value**

0x0

**Description**

The value of SMIDR\_EL1.SH.

**sve.sve2\_version****Type**

int

**Default value**

0x0

**Description**

The version of SVE2 if implemented. Possible values are: 0 - FEAT\_SVE2, 1 - FEAT\_SVE2p1, 2 - FEAT\_SVE2p2.

**sve.sve\_dabt\_far\_behaviour****Type**

int

**Default value**

0x0

**Description**

Whether the FAR reported on a Data Abort is imprecise. Possible values are: 0 - FAR Precise, 1 - FAR not Precise on abort due to an SVE contiguous vector load/store in Streaming SVE mode, or an SME load/store. , 2 - As per 1, but only for predicated SVE/SME instructions.

**sve.sve\_wp\_far\_behaviour****Type**

int

**Default value**

0x0

**Description**

FAR reporting behavior on a Watchpoint debug exception. Possible values are: 0 - FAR Precise, 1 - FAR not Precise on abort due to an SVE contiguous vector load/store in Streaming SVE mode, or an SME load/store, 2 - FAR not valid on abort due to an SVE contiguous vector load/store in Streaming SVE mode, or an SME load/store, 3 - As per 1, but only for predicated SVE/SME instructions, 4 - As per 1, but only for predicated SME/SVE load/store instructions that are executed in Streaming Mode.

**sve.trace\_za\_tilewise****Type**

bool

**Default value**

0x1

**Description**

Whether tile-wise accesses to ZA are traced tile-wise rather than array-wise. Note: if false, column-wise accesses cause an event for every vector in the tile.

**sve.undef\_invalid\_combined\_movprfx****Type**

bool

**Default value**

0x1

**Description**

If a combined MOVPRFX is invalid, raise an UNDEF exception. Otherwise, NOP the second half. This parameter is deprecated.

**sve.unknown\_value****Type**

int

**Default value**

0xdeaddeaddeaddead

**Description**

Simulated value for a state that has an UNKNOWN value after reset.

**sve.vecLEN****Type**

int

**Default value**

0x8

**Description**

SVE vector length in units of 64 bits.

**sve.z\_reg\_on\_load\_fault\_behaviour****Type**

int

**Default value**

0x0

**Description**

Governs the behavior of destination Z-registers in case of a load fault. Possible values are: 0 - Register becomes UNKNOWN, 1 - Register is preserved.

**sve.za\_on\_svl\_increase\_behaviour****Type**

int

**Default value**

0x0

**Description**

Controls the state of the previously inaccessible portion of the ZA registers on SVL increase. Possible values are: 0 - Retain values, 1 - Zero ZA.

**sve.za\_tag\_checked****Type**

bool

**Default value**

0x1

**Description**

Whether memory accesses due to SME LDR and STR instructions that access the SME ZA array perform a Tag Check.

**swp\_with\_xzr\_is\_st\_atomic****Type**

bool

**Default value**

0x1

**Description**

If true, swp with dest as xzr is treated as store atomic.

**sync\_ext\_abort\_is\_sync\_serror****Type**

bool

**Default value**

0x0

**Description**

Treat synchronous external aborts as synchronous SErrors (RASv8.9). 0, synchronous external abort. 1, synchronous serror.

**system\_pmu\_id****Type**

int

**Default value**

0x0

**Description**

When FEAT\_SPMU is implemented, indicates the largest value *s* to select a System PMU <*s*>.

**take\_ccfail\_tsc\_trap****Type**

bool

**Default value**

0x0

**Description**

When take\_ccfail\_undef=1 this parameter controls whether or not an SMC instruction that is trapped by HCR\_EL2.TSC but fails its condition code check generates a trap to EL2.

**take\_ccfail\_undef****Type**

bool

**Default value**

0x1

**Description**

UNDEF exception is taken even if condition code check fails.

**tcr\_ps\_reserved\_value\_size****Type**

int

**Default value**

0x0

**Description**

Physical size treated when TCR.(I)PS is programmed with a reserved value. 0, 48 bits. 1, 52 bits. The parameter value is treated 0 if LPA is not supported.

**tcr\_txsz\_undersize\_should\_fault****Type**

bool

**Default value**

0x1

**Description**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

**tdosa\_traps\_osdlr\_if\_no\_os\_double\_lock****Type**

bool

**Default value**

0x1

**Description**

MDCR\_EL\*.TDOSA enables trap on OSDLR\_EL1 and DBGOSDLR when OS double-lock is not implemented.

**tidcp\_traps\_el0\_undef\_imp\_def****Type**

bool

**Default value**

0x1



**Description**

TIDCP has priority over UNDEF for accesses to IMPLEMENTATION DEFINED functionality from ELO.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_or\_ic\_invalid\_xt****Type**

int

**Default value**

0x0

**Description**

Behavior of TLBI and IC instructions that don't take Xt as an argument when Xt != 0b11111.  
0: TLB and IC not UNDEF, 1: TLBI UNDEF, IC not UNDEF, 2: TLBI not UNDEF, IC UNDEF, 3: TLBI and IC UNDEF.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**tme\_disable-read-write-set-optimizations****Type**

bool

**Default value**

0x0

**Description**

If true, disables the read/write set related optimizations of the model.

**tme\_imp-failures-can-retry****Type**

bool

**Default value**

0x0

**Description**

If true, IMP=1 failures introduced by the parameters: tme\_wfe-fails-transactions tme\_tcommit-fails-transactions tme\_wakeup-from-wfe-always-fails-transactions will also report RTRY=1.

**tme\_implementation-type****Type**

int

**Default value**

0x0

**Description**

Implementation type for TME. The following options are available: 0x0: Always fail starting transactions with the IMP cause. 0x1: Fail on forbidden operations (e.g. some system register accesses) and at model's convenience. 0x2: As 0x1 but also enable inter PE memory conflict checking. .

**tme\_random-memory-access-fail-chance****Type**

int

**Default value**

0x0

**Description**

If >0, add a pseudorandom chance for every memory access (loads, stores, TCOMMIT) inside a transaction to cause the transaction to fail with IMP.

**tme\_read-set-size****Type**

int

**Default value**

0x0

**Description**

Size of the transactional read set in bytes, rounded up to the nearest integer number of transaction granules. 0 == unlimited.

**tme\_support-only-guaranteed-mem-attr****Type**

bool

**Default value**

0x0

**Description**

If true, a transactional access to memory with a type not architecturally guaranteed to be supported will cause a transaction failure with IMP=1.

**tme\_tcommit-fails-transactions****Type**

bool

**Default value**

0x0

**Description**

If true, executing TCOMMIT inside a transaction will cause it to fail with IMP=1.

**tme\_wakeup-from-wfe-always-fails-transactions****Type**

bool

**Default value**

0x0

**Description**

If true, waking up from a WFE will always fail the transaction, even if not required.

**tme\_wfe-fails-transactions****Type**

bool

**Default value**

0x0

**Description**

If true, executing WFE inside a transaction will cause it to fail with IMP=1.

**tme\_write-set-size****Type**

int

**Default value**

0x0

**Description**

Size of the transactional write set in bytes, rounded up to the nearest integer number of transaction granules. 0 == unlimited.

**`trace_full_simd_reg_with_nep`****Type**

bool

**Default value**

0x0

**Description**

Whether full simd register is traced even if partial update is done when FPCR.NEP=1.

**`trace_has_sysreg_access`****Type**

bool

**Default value**

0x1

**Description**

ETM trace registers support access via system registers.

**`trace_icc_registers_as_icv_when_redirected`****Type**

bool

**Default value**

0x0

**Description**

If true, update trace with ICV, instead of ICC when ICV registers are accessed depending on the core state.

**`trace_physical_registers_when_host_virtualisation_enabled`****Type**

int

**Default value**

0x0

**Description**

When host virtualisation is enabled, trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers.

**trace\_xzr\_in\_core\_regs64\_trace****Type**

bool

**Default value**

0x1

**Description**

Whether CORE\_REGS64\_READ traces XZR and WZR input registers.

**trap\_dc\_cmo\_to\_pou\_if\_nop****Type**

bool

**Default value**

0x1

**Description**

Whether traps to DC CMO operations to PoU are ignored if the same is treated as NOP.

**trap\_ic\_cmo\_to\_pou\_if\_nop****Type**

bool

**Default value**

0x1

**Description**

Whether traps to IC CMO operations to PoU are ignored if the same is treated as NOP.

**trap\_reserved\_group3\_id\_regs****Type**

bool

**Default value**

0x0

**Description**

Whether setting HCR\_EL2.TID3 traps reserved group3 id registers.

**trbe\_cmod****Type**

int

**Default value**

0x0

**Description**

TRBE Customer Modified.

**trbe\_des****Type**

int

**Default value**

0x0

**Description**

Designer, JEP106 identification code.

**trbe\_external\_abort\_handling****Type**

int

**Default value**

0x0

**Description**

Describes how the PE manages External aborts on writes made by the Trace Buffer Unit to the trace buffer. (0->External abort is reported to TRBE. From Armv9.3, the value 0 is not permitted and will be 1 if Armv9.3 is implemented. 1-> External abort is ignored. 2->The External abort generates an SError and the error is not reported to TRBE.).

**trbe\_has\_hardware\_translation\_table\_update****Type**

bool

**Default value**

0x1

**Description**

If true, address translation performed by the Trace Buffer Extension manages the Access Flag and dirty state.

**trbe\_implemented\_version****Type**

int

**Default value**

0x1

**Description**

Trace Buffer Extension implemented version, 1: FEAT\_TRBE implemented (Armv9.0), 2: FEAT\_TRBEv1p1 and FEAT\_TRBE\_EXC are implemented.

**trbe\_mpam****Type**

int

**Default value**

0x0

**Description**

TRBE MPAM support.

**trbe\_part****Type**

int

**Default value**

0x0

**Description**

Part number.

**trbe\_partid\_max****Type**

int

**Default value**

0x0

**Description**

Largest permitted TRBDEVID1.PARTID value.

**trbe\_pmg\_max****Type**

int

**Default value**

0x0

**Description**

Largest permitted TRBDEVID1.PMG value.

**trbe\_revand****Type**

int

**Default value**

0x0

**Description**

TRBE component minor revision.

**trbe\_revision****Type**

int

**Default value**

0x0

**Description**

TRBE architecture revision.

**trbe\_stop\_on\_misaligned\_pointers****Type**

bool

**Default value**

0x0

**Description**

If true, the Trace Buffer Extension will stop tracing if a buffer pointer is not aligned to TRBIDR\_EL1.Align.

**treat-dcache-cmos-to-occ-as-nop****Type**

bool

**Default value**

0x0

**Description**

Implement CMOs to Outer cache level as NOP.

**treat-dcache-cmos-to-poc-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache maintenance operations to the point of coherency are required for instruction to data coherence. 0 - Clean/Invalidate ops required, 1 - Clean/Invalidate ops not required and cannot generate faults, 2 - Clean/Invalidate ops not required but can generate faults.

**treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0



**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**`treat-dcache-invalidate-as-clean-invalidate`****Type**

bool

**Default value**

0x0

**Description**

Treat data cache invalidate operations as clean and invalidate.

**`treat-icache-cmos-to-pou-as-nop`****Type**

int

**Default value**

0x0

**Description**

If `has_coherent_icache` is true, whether instruction cache invalidation operations to PoU which are treated as NOP can generate fault. 0 - cannot generate faults, 1 - can generate faults.

**`treat_PAC_as_NOP`****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**`treat_forced_normal_as_device_for_excl_atomics`****Type**

bool

**Default value**

0x0

**Description**

Whether exclusive/atomic access is supported in same manner as access to device if stage1 is Device memory and final memory attribute forced to normal by FWB.

**`treat_pld_as_nop`****Type**

bool

**Default value**

0x0

**Description**

If true, treat PLD as NOP.

**`treat_pli_as_nop`****Type**

bool

**Default value**

0x0

**Description**

If true, treat PLI as NOP.

**`treat_wfi_wfe_as_nop`****Type**

bool

**Default value**

0x0

**Description**

If true, never go into wait state for WFI or WFE instructions.

**`truncate_pc_on_illegal_exception_return_to_aarch32`****Type**

bool

**Default value**

0x1

**Description**

On Illegal ERET to AArch32, truncate PC to 32-bits.

**`tune_spe_cache_events`****Type**

string

**Default value**

""

**Description**

"Set the percentage of cache event bits set on a load instruction. Each tuning must be between 5 and 95. JSON schema for the parameter value is: {'l2\_dcache\_access':20, 'l2\_dcache\_miss':20, 'dcache\_modified':20, 'recently\_fetched':20, 'data\_snooped':20}."

**undef\_ccsidr2\_access\_for\_unimplemented\_aarch32****Type**

bool

**Default value**

0x0

**Description**

Whether access to CCSIDR2 is undef if AArch32 is implemented or not at EL1.

**unification-level****Type**

int

**Default value**

0x1

**Description**

Level of Unification Inner Shareable for the cache hierarchy.

**unification-uniprocessor-level****Type**

int

**Default value**

0x1

**Description**

Level of Unification Uniprocessor for the cache hierarchy.

**unpred\_LSE128\_overlap****Type**

int

**Default value**

0x1

**Description**

Constrained unpredictable behaviours for 128-bit LSE overlap. 1 Constraint\_UNDEF, 2 Constraint\_NOP.

**unpred\_brb\_iall\_or\_inj\_invalid\_xt\_behave\_as\_undef****Type**

bool

**Default value**

0x0

**Description**

If true, BRB IALL/INJ instruction will behave as UNDEFINED if Xt != 0b11111.

**unpred\_brbe\_next\_branch\_cycle\_count\_unknown****Type**

bool

**Default value**

0x0

**Description**

If true, cycle count value for the next BRBE branch record after BRB INJ execution outside prohibited region is unknown.

**unpred\_clear\_ISV\_for\_exception\_before\_software\_step****Type**

bool

**Default value**

0x0

**Description**

Whether ESR\_ELx.ISV bit is cleared/set, when it is constrained unpredictable due to a different exception before a software step exception.

**unpred\_edscr\_ns\_set\_unknown\_bit****Type**

bool

**Default value**

0x0

**Description**

Unknown(x) bit in NS field in EDSCR can be configure to 0 or 1.

**unpred\_edscr\_rw\_unknown\_bits\_read\_as\_1****Type**

bool

**Default value**

0x0

**Description**

Unknown(x) bits in RW field in EDSCR are read as 1 instead of 0.

**unpred\_edscr\_status\_read\_as\_no\_syndrome****Type**

bool

**Default value**

0x0

**Description**

Controls the choice of EDSCR.STATUS bit-values, when it is constrained unpredictable behaviour due to a different exception before a halting step debug event.

**unpred\_extdbg\_unknown\_bits****Type**

int

**Default value**

0x0

**Description**

Data used to fill only in UNKNOWN bit-fields of external debug registers e.g., EDPFR and EDDFR.

**unpred\_load\_single\_reg\_overlap\_with\_wb****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable behaviours for single load with writeback(might impact certain load pair instructions) 0 Constraint\_WBSUPPRESS, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

**unpred\_mrsmsr\_currentlymapped\_undef****Type**

bool

**Default value**

0x0

**Description**

UNPREDICTABLE register access (accessible from current mode using different instruction) modeled as NOP when false and UNDEF when true.

**unpred\_mrsmsr\_protfailed\_undef****Type**

bool

**Default value**

0x0

**Description**

UNPREDICTABLE register access (not accessible from current PL and security state) modeled as NOP when false and UNDEF when true.

**unpred\_mte\_stzgm\_tag\_operation\_before\_data****Type**

bool

**Default value**

0x1

**Description**

Whether Tag operations are performed before data operations for an STGZM instruction.

**unpred\_mte\_tag\_read\_when\_ata\_controls\_are\_zero\_or\_untagged\_attr****Type**

bool

**Default value**

0x0

**Description**

Constrained unpredictable for MTE tag read when ATA controls are 0 or untagged attribute. false, Read as zero. true, Permitted to generate an external abort if a read of data from the same address would generate an external abort.

**unpred\_mte\_tag\_store\_data\_cache\_instr\_to\_device\_mem\_as\_alignment\_fault****Type**

bool

**Default value**

0x0

**Description**

Constrained unpredictable choice for MTE instructions which store tags (on DC instructions) to memory locations marked as Device. false, Storing the data, if any, to the locations. true, Generating an Alignment Fault.

**unpred\_mte\_tag\_store\_to\_device\_mem\_as\_alignment\_fault****Type**

bool

**Default value**

0x0

**Description**

Constrained unpredictable choice for STZGM instruction which store tags to memory locations marked as Device. false, Storing the data, if any, to the locations. true, Generating an Alignment Fault.

**unpred\_nested\_virtualization\_nv\_behaviour****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable choices for HCR\_EL2.NV=0 and HCR\_EL2.NV1=1 with respect to nested virtualization - 0, Behave as defined in the specification as per bit values - 1, Behave as if HCR\_EL2.NV=1 and HCR\_EL2.NV1=1 for all purpose other than reading back HCR\_EL2.NV - 2, Behave as if HCR\_EL2.NV=0 and HCR\_EL2.NV1=0 for all purpose other than reading back HCR\_EL2.NV1 .

**unpred\_par\_attr\_returns\_mair****Type**

bool

**Default value**

0x0

**Description**

If true, PAR\_EL1.ATTR represents the memory attributes as per the MAIR value instead of the ones in the descriptor.

**unpred\_sctlr\_c\_0\_taggable\_behaviour****Type**

int

**Default value**

0x2

**Description**

Controls unpredictable effects when SCTLTR\_ELx.C=0 for a stage 1 translation regime on whether memory is treated as taggable. Values: 0=Tagged, 1=Untagged but forced to Tagged when FWB=1 and stage 2 restores WB, 2 = Untagged.

**unpred\_store\_exclusive\_base\_overlap****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable behaviours for store exclusive when s==n. 0 Constraint\_NONE, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

**unpred\_store\_pair\_and\_single\_reg\_overlap\_with\_wb****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable behaviours for pair and single store with writeback(doesn't cover store exclusive) 0 Constraint\_NONE, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

**unpred\_tlbi\_not\_in\_monitor\_mode****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable behaviors for AArch32 TLBI instructions executed in secure privileged mode other than Monitor mode. 0: Preferred behavior (default), 1: UNDEF, 2: NOP, 3: execute as if had been executed in Monitor mode.

**unpred\_tsize\_aborts****Type**

bool

**Default value**

0x0

**Description**

Behaviour when TSize is out of range. 0, force into range. 1, translation fault, forces unpred\_tsize\_pamax\_aborts to 1.

**unpred\_tsize\_pamax\_aborts****Type**

bool

**Default value**

0x0



**Description**

Behaviour when stage 2 TSize exceeds the physical address size (or 40bits, from AArch32). 0, force into range. 1, translation fault. Ignored if unpred\_tsize\_aborts is 1.

**unpred\_vnchr\_el2\_ress\_mismatch****Type**

int

**Default value**

0x0

**Description**

Constrained unpredictable choices when bits marked as RESS do not all have the same value for VNCR\_EL2 - 0, Generating an EL2 translation regime translation abort on use of the VNCR\_EL2 register - 1, Reserved sign extended bits of VNCR\_EL2 are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes other than reading back the register - 2, Reserved sign extended bits of VNCR\_EL2 are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes .

**unpred\_zero\_spsr\_btype****Type**

bool

**Default value**

0x1

**Description**

Constrained unpredictable control to make SPSR\_ELx.BTYPE 0 instead of PSTATE.BTYPE on synchronous exceptions other than Software Step, PC alignment fault, Instruction Abort, Breakpoint or Address Matching Vector Catch, Illegal Execution State, BRK instruction, Branch Target.

**unpredictable\_exclusive\_abort\_memtype****Type**

int

**Default value**

0x0

**Description**

Cause MMU abort if exclusive access is not supported in certain memory type (0=exclusives allowed in all memory types, 1=exclusives abort in Device memory types, 2=exclusives abort in any type other than WB inner cacheable).

**unpredictable\_hvc\_behaviour****Type**

int

**Default value**

0x0

**Description**

HVC unpredictable behaviour. 0, UNDEF. 1, NOP.

**unpredictable\_smc\_behaviour****Type**

int

**Default value**

0x0

**Description**

SMC unpredictable behaviour. 0, UNDEF. 1, NOP.

**unpredictable\_wfet\_and\_wfit\_behaviour****Type**

int

**Default value**

0x1

**Description**

WFET and WFIT unpredictable behaviour in debug state. 0, UNDEFINED. 1, NOP.

**unsupported\_atomic\_fault\_type****Type**

int

**Default value**

0x0

**Description**

Type of fault reported on unsupported atomic access. 0 = external abort if any reported by interconnect, 1 = precise unsupported atomic fault, 2 = precise external abort, 3 = imprecise external abort.

**unsupported\_fp8\_format\_behaviour****Type**

int

**Default value**

0x0

**Description**

Behaviour when FPMR.{F8S1,F8S2,F8D} are programmed to a reserved value 0->FP8 Inputs are treated as a signalling NaN, FP8 outputs are 0xFF 1->Format is treated as FPMR.{F8S1,F8S2,F8D} & 0x1.

**unsupported\_hw\_update\_fault\_type****Type**

int

**Default value**

0x0

**Description**

Type of abort reported when hw update to descriptor is done using unsupported memtype (0=No abort, 1=IMPDEF abort caused by memtype, 2=Sync external abort).

**use\_architectural\_names****Type**

bool

**Default value**

0x0

**Description**

Use names SP/LR/PC instead of R13/R14/R15.

**use\_rosetta\_disass****Type**

int

**Default value**

0x1

**Description**

Use Rosetta disassembly library. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**use\_sif\_to\_compute\_pan****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_PAN3 is implemented, whether SCR\_EL3.SIF bit is used to determine instruction access permission for the purpose of PAN.

**use\_stage1\_sh\_as\_input\_to\_stage2****Type**

bool

**Default value**

0x0

**Description**

IMPDEF case of whether to use stage1 shareability or OuterShareable as input to stage2 if stage1 is Device memtype.

**use\_tlb\_contig\_hint****Type**

bool

**Default value**

0x0

**Description**

Translation table entries with the contiguous hint bit set generate large TLB entries.

**user\_defined\_rom\_table\_debug\_power\_config****Type**

string

**Default value**

""

**Description**

User defined ROM Table debug power domains for ED,CTI,PMU and TRACE, and DBGPCR configuration. The "version" field and "cores" array are mandatory. The "dbgpcr" array, if provided, must contain unique integers in the range [0, 32) describing which debug power domains have power control implemented. The "rom" and "dbgpcr" fields in objects in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. All power domain ID fields ("rom", "ed/pmu", "cti", "etm") must be in the range [0, 32). The "ed/pmu" field is mandatory. Example JSON for a hierarchical debug ROM layout: '{"version": 0, "dbgpcr": [0, 1], "cores": [{"dbgpcr": [1, 31], "rom": 0, "ed/pmu": 0, "cti": 31, "etm": 1}, {"ed/pmu": 0}]}'.

**vpu\_datapath\_width****Type**

int

**Default value**

0x80

**Description**

VPU data path width.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**warn\_for\_dbgwcr\_reserved\_values\_with\_razwi\_bits****Type**

bool

**Default value**

0x1

**Description**

Display a warning when DBGWCR is programmed with a reserved value even if some bits(e.g. HMC) are RAZ/WI.

**warn\_unpredictable\_in\_v7****Type**

bool

**Default value**

0x1

**Description**

If true, behaviour which is unpredictable in V7 yet is predictable in V8 will produce a warning.

**watchpoint-log2secondary\_restriction****Type**

int

**Default value**

0x0

**Description**

log2 size of secondary restriction of FAR/EDWAR possible values on watchpoint hit for load/store operations.

**wfe\_wakeup\_delay****Type**

int

**Default value**

0x0

**Description**

Configure WFE wakeup delay in CPU cycles.

**wfi\_wakeup\_delay****Type**

int

**Default value**

0x0

**Description**

Configure WFI wakeup delay in CPU cycles.

**`wnr_is_read_for_s2f_on_s1_atomic_instr_fault`****Type**

bool

**Default value**

0x0

**Description**

Whether WnR is 0 for stage2 fault on stage1 for atomic instructions.

**`wnr_is_read_for_s2f_on_s1_dbm_update`****Type**

bool

**Default value**

0x0

**Description**

Whether WnR is 0 for stage2 fault on stage1 descriptor dbm update.

**`wp_ignores_dbm_update`****Type**

bool

**Default value**

0x0

**Description**

If true, dbm update is ignored on watchpoint hit.

**`wp_num_reporting`****Type**

int

**Default value**

0x0

**Description**

When reporting of the watchpoint number on Watchpoint Exceptions and Debug Events is performed 0 - When FEAT\_Debugv8p9 is implemented or otherwise required 1 - When FEAT\_Debugv8p9 or FEAT\_SME is implemented.

### 3.5.3 ARMAEMv8MCT

ARMAEMv8MCT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-191: IP revisions support**

Revision	Quality level
v8.0M	Full support
v8.1M	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Iris and MTI instances for ARMAEMv8MCT

This model has the following Iris instances:

**Table 3-192: ARMAEMv8MCT Iris instances**

InstanceName	ComponentName
ARMAEMv8MCT	ARM_AEMv8M
ARMAEMv8MCT.acp_mapper	PVBusMapper
ARMAEMv8MCT.ext_bus	PVBusLogger
ARMAEMv8MCT.ext_bus.mapper	PVBusMapper
ARMAEMv8MCT.l1_incoherent_interconnect	PVCache
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMAEMv8MCT.l1dcache	PVCache

InstanceName	ComponentName
ARMAEMv8MCT.l1dcache.upstream[0]	PVBusSlave
ARMAEMv8MCT.l1icache	PVCache
ARMAEMv8MCT.l1icache.upstream[0]	PVBusSlave
ARMAEMv8MCT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-193: ARMAEMv8MCT MTI instances**

InstanceName	ComponentName
ARMAEMv8MCT	ARM_AEMv8M
ARMAEMv8MCT.acp_mapper	PVBusMapper
ARMAEMv8MCT.ext_bus	PVBusLogger
ARMAEMv8MCT.ext_bus.mapper	PVBusMapper
ARMAEMv8MCT.l1_incoherent_interconnect	PVCache
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMAEMv8MCT.l1dcache	PVCache
ARMAEMv8MCT.l1dcache.upstream[0]	PVBusSlave
ARMAEMv8MCT.l1icache	PVCache
ARMAEMv8MCT.l1icache.upstream[0]	PVBusSlave
ARMAEMv8MCT.l2_flusher	AsyncCacheFlushUnit



## Ports for ARMAEMv8MCT

Table 3-194: Ports

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
ahbp_m	PVBus	Master	The core will generate Vendor System data accesses on this port.
ahbs	PVBus	Slave	External master (e.g. DMA) can write TCMs (whether or not enabled in xTCMCR).
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
coproc_bus	CoprocBusProtocol	Slave	Co-Processor Interface
cpuwait	Signal	Slave	When this signal is HIGH out of reset, it forces the processor into a quiescent state that delays its boot-up sequence and instruction execution until this signal is driven LOW.
currpri	Value	Master	Current execution priority.
dap_s	PVBus	Slave	Debug Access Port (DAP).
dbgen	Signal	Slave	Invasive debug enable.
dbgrestart	Signal	Slave	External debug request.
dbgrestarted	Signal	Master	External debug request.
edbgrq	Signal	Slave	External debug request.
etm_reset	Signal	Slave	Separate reset for ETM, if param "has_etm_reset" is true.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	Value	Master	Port that sends the value of the FPxxC exception flags (FPIXC, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	Signal	Master	External debug request.
idau	PVBus	Master	The core will generate IDAU Bus request.
idau_invalidate_region	Value_64	Slave	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
initpahben	Signal	Slave	Enable P-AHB on the next reset
initvtor_ns	Value	Slave	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initvtor_s	Value	Slave	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
intisr[496]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.

Name	Protocol	Type	Description
intnum	Value	Master	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
io_port_in	PVBus	Slave	I/O port pair. See the documentation for the io_port_out port.
io_port_out	PVBus	Master	I/O port pair. Used if IOP is true. Transactions from io_port_out which do not "match" should be returned via io_port_in. For performance reasons, the I/O port interface is not modelled directly. Instead, a simple PVBus gasket is inserted at the point in the memory system where the I/O port would be. In hardware, a device would be attached to the port which would tell the CPU whether it would like to intercept each transaction, given its address. This can be modelled in a performant manner by connecting a PVBusMapper-based device to io_port_out which intercepts transactions of interest and passes other transactions back to the CPU via io_port_in. Your I/O port device model is also responsible for aborting transactions which would be aborted on hardware (e.g. exclusives) if necessary.
lockdcaic	Signal	Slave	-
locknsmpu	Signal	Slave	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	Signal	Slave	Disable writes to VTOR_NS
lockpahb	Signal	Slave	P-AHB related ports Disable writes to PAHBCCR
locksau	Signal	Slave	Disable writes to the SAU_* registers
locksmpu	Signal	Slave	Disable writes to the Secure MPU_* registers
locksvtaircr	Signal	Slave	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
lockup	Signal	Master	Asserted when the processor is in lockup state.
niden	Signal	Slave	Non-invasive debug enable.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
spiden	Signal	Slave	Secure invasive debug enable.
spniden	Signal	Slave	Secure non-invasive debug enable.
stcalib[2]	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARMAEMv8MCT

### AFSR\_type

#### Type

int

**Default value**

0x1

**Description**

0:No AFSR, 1:state-only 2:Many bits set from (im)precise aborts on AXI, TCM, etc.

**AIRCR.BFHFNMINReset****Type**

bool

**Default value**

0x0

**Description**

If true, set the bit after reset (as if by IMP\_DEF mechanism). Ignored if SECEXT=false.

**AIRCR.BFHFNMINWritable****Type**

bool

**Default value**

0x1

**Description**

Is AIRCR.BFHFNMIN bit[13] writeable.

**AIRCR.ENDIANNESS****Type**

bool

**Default value**

0x0

**Description**

Initialize processor to big endian mode.

**AIRCR.PRISWritable****Type**

bool

**Default value**

0x1

**Description**

Is AIRCR.PRIS bit[14] writeable.

**AIRCR.VECTCLRACTIVEChangesMode****Type**

bool

**Default value**

0x1

**Description**

Asserting AIRCR.VECTCLRACTIVE clears IPSR and any active exceptions. The mode is also changed to thread if this flag is true. Ignored for v8-M.

**AIRCR\_NS.DIT\_reset****Type**

bool

**Default value**

0x1

**Description**

If true and AIRCR\_NS.DIT\_writable==0, set the bit after reset (as if by IMP\_DEF mechanism).

**AIRCR\_NS.DIT\_writable****Type**

bool

**Default value**

0x1

**Description**

Is AIRCR\_NS.DIT bit[4] writeable.

**AIRCR\_S.DIT\_reset****Type**

bool

**Default value**

0x1

**Description**

If true and AIRCR\_S.DIT\_writable==0, set the bit after reset (as if by IMP\_DEF mechanism).

**AIRCR\_S.DIT\_writable****Type**

bool

**Default value**

0x1

**Description**

Is AIRCR\_S.DIT bit[4] writeable.

**BB\_PRESENT****Type**

bool

**Default value**

0x0

**Description**

Enable bitbanding.

**BEATS\_PER\_TICK****Type**

int

**Default value**

0x2

**Description**

Number of beats from each in-flight vector instruction executed in 1 tick (1,2 or 4).

**BF\_is\_nop****Type**

bool

**Default value**

0x1

**Description**

BF instruction executes as NOP, even if we have LO\_BRANCH\_INFO.

**CCR.BP****Type**

bool

**Default value**

0x1

**Description**

Reset value of the Configuration and Control Register's branch prediction enable bit.

**CCR.BP\_writable****Type**

bool

**Default value**

0x0

**Description**

Whether it is possible to modify the Configuration and Control Register's branch prediction enable bit.

**CDEMAPPEDONCP****Type**

int

**Default value**

0xff

**Description**

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

**CDERTLID****Type**

int

**Default value**

0x20

**Description**

Value of ID\_AFR0.CDERTLID.

**CFGMEMALIAS****Type**

int

**Default value**

0x0

**Description**

Memory address alias bit for the ITCM, DTCM and P-AHB regions. 0=No alias, 1=Alias bit 24, 2=Alias bit 25, 4=Alias bit 26, 8=Alias bit 27, 16=Alias bit 28.

**CFGNOCDECP****Type**

int

**Default value**

0x0

**Description**

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

**CFGPAHBSZ****Type**

int

**Default value**

0x0

**Description**

Size of the P-AHB peripheral port memory region. 0=P-AHB disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

**CPNSPRESENT****Type**

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

**CPSPRESENT****Type**

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

**CPUID****Type**

int

**Default value**

0x0

**Description**

Set SCS CPUID Base Register. If set to zero, a default CPUID is used.

**CTI****Type**

bool

**Default value**

0x0

**Description**

CTI (Cross Trigger Interface) included.

**CTI\_irq0\_pin****Type**

int

**Default value**

0x4

**Description**

CTI interrupt request 0 pin.

**CTI\_irq1\_pin****Type**

int

**Default value**

0x5

**Description**

CTI interrupt request 1 pin.

**DTGU****Type**

bool

**Default value**

0x0

**Description**

DTCM Security Gate Unit included.

**DTGUBLKSZ****Type**

int

**Default value**

0x3

**Description**

DTCM gate unit block size. Size=pow(2, DTGUBLKSZ + 5) bytes.

**DTGUMAXBLKS****Type**

int

**Default value**

0x0



**Description**

Maximum number of DTCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{DTGUMAXBLKS})$ .

**DWT\_CTRL.NOCYCCNT****Type**

bool

**Default value**

0x0

**Description**

DWT cycle-counter not present (v8M-bl/v6M never have one).

**DWT\_CTRL.NOPRFCNT****Type**

bool

**Default value**

0x0

**Description**

DWT performance-counters not present (v8M-bl/v6M never have them).

**DWT\_CTRL.NUMCOMP****Type**

int

**Default value**

0x4

**Description**

Number of watchpoint unit comparators implemented.

**DWT\_DEVARCH.REVISION****Type**

int

**Default value**

0x1

**Description**

0: V2, 1: V2.1.

**DWT\_FUNCTION0.ID****Type**

int

**Default value**

0xb

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION0. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION1.ID****Type**

int

**Default value**

0x1e

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION1. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION10.ID****Type**

int

**Default value**

0xb

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION10. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION11.ID****Type**

int

**Default value**

0x1e

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION11. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION12.ID****Type**

int

**Default value**

0xb

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION12. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION13.ID****Type**

int

**Default value**

0x1e

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION13. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION14.ID****Type**

int

**Default value**

0xb

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION14. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION15.ID****Type**

int

**Default value**

0x1e

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION15. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION2.ID****Type**

int

**Default value**

0xb

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION2. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION3.ID****Type**

int

**Default value**

0x1e

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION3.  
If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION4.ID****Type**

int

**Default value**

0xb

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION4.  
If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION5.ID****Type**

int

**Default value**

0x1e

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION5.  
If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION6.ID****Type**

int

**Default value**

0xb

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION6.  
If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION7.ID****Type**

int

**Default value**

0x1e

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION7. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION8.ID****Type**

int

**Default value**

0xb

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION8. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_FUNCTION9.ID****Type**

int

**Default value**

0x1e

**Description**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION9. If 'baseline' is set, invalid ID bits are cleared.

**DWT\_TRACE****Type**

bool

**Default value**

0x1

**Description**

Support for DWT trace, controls the DWT\_CTRL.NOTRCPKT bit. false : No DWT trace included, true: DWT trace included.

**DWT\_VMASK\_reset\_data****Type**

int

**Default value**

0x0

**Description**

DWT\_VMASK register reset value.

**ERRDEVID.NUM****Type**

int

**Default value**

0x38

**Description**

RAS: Number of implemented error record indexes, 0 to 56.

**FPB\_HAS\_LSR****Type**

bool

**Default value**

0x1

**Description**

FPB has LAR and LSR for software lock if mainline.

**FP\_CTRL.NUM\_CODE****Type**

int

**Default value**

0x8

**Description**

Number of breakpoint unit comparators implemented (limited to 15 in V6M or baseline).

**FP\_CTRL.NUM\_LIT****Type**

int

**Default value**

0x0

**Description**

How many Literals FPB supports remapping (ignored if baseline or TZM).

**FP\_REMAP.RMPSPT****Type**

bool

**Default value**

0x1

**Description**

FPB supports remapping (ignored if baseline or SECEXT).

**ID\_DFR0.Debug\_Model\_M\_profile****Type**

bool

**Default value**

0x1

**Description**

Set whether debug extensions are implemented.

**ID\_ISAR0.CmpBranch****Type**

int

**Default value**

0x3

**Description**

Support for Compare and Branch instructions. 1 = Supports CBNZ and CBZ instructions; 3 = Supports non-predicated low overhead looping (WLS, DLS, LE, and LC) and branch future (BF, BFX, BFL, BFLX, and BFCSEL) instructions.

**ID\_ISAR0.coproc\_instrs****Type**

int

**Default value**

0x4

**Description**

Supported Coprocessor instructions 0: None 1: CDP, LDC, MCR, MRC, and STC instructions 2: As for 1, and CDP2, LDC2, MCR2, MRC2, and STC2 instructions 3: As for 2, and MCRR and MRRC instructions 4: As for 2, and MCRR and MRRC instructions.

**ID\_ISAR1.extend\_instrs****Type**

int

**Default value**

0x2

**Description**

level of support for extend instructions.

**ID\_ISAR1.interwork\_instrs****Type**

int

**Default value**

0x2

**Description**

level of support for Interworking instructions.

**ID\_ISAR2.MultiAccessInt****Type**

int

**Default value**

0x2

**Description**

level of support for interruptible multi-access instructions.

**ID\_ISAR2.multS\_instrs****Type**

int

**Default value**

0x3

**Description**

level of support for advanced signed Multiply instructions.

**ID\_ISAR2.multU\_instrs****Type**

int

**Default value**

0x2

**Description**

level of support for advanced unsigned Multiply instructions.

**ID\_ISAR3.SIMD\_instrs****Type**

int

**Default value**

0x3

**Description**

level of support for SIMD instructions.

**ID\_ISAR3.saturate\_instrs****Type**

int



**Default value**

0x1

**Description**

level of support for saturate instructions.

**ID\_ISAR3.synchprim\_instrs****Type**

int

**Default value**

0x1

**Description**

level of support for synchronization primitives ID\_ISAR3.

**ID\_ISAR4.synchPrim\_instrs\_frac****Type**

int

**Default value**

0x3

**Description**

level of support for synchronization primitives ID\_ISAR4.

**ID\_ISAR4.unpriv\_instrs****Type**

int

**Default value**

0x2

**Description**supported unprivileged instructions 0: None 1: LDRBT, LDRT, STRBT, and STRT instructions  
2: As for 1, and LDRHT, LDRSBT, LDRSHT, and STRHT instructions.**ID\_ISAR4.withshifts\_instrs****Type**

int

**Default value**

0x3

**Description**

level of support for instructions with shifts.

**ID\_ISAR5.PACBTI****Type**

int

**Default value**

0x0

**Description**

0: PAC/BTI not implemented, 1: PAC implemented using the QARMA5 algorithm with BTI, 2: PAC implemented using an IMP DEF algorithm with BTI, 4: PAC implemented using the QARMA3 algorithm with BTI.

**ID\_MMFR0.Auxiliary\_registers****Type**

bool

**Default value**

0x1

**Description**

Auxiliary registers bits in ID\_MMFR0, indicate the support for Auxiliary registers.

**ID\_MMFR0.Outermost\_shareability****Type**

int

**Default value**

0x0

**Description**

Outermost shareability bits in ID\_MMFR0, indicate the outermost shareability domain implemented.

**ID\_MMFR0.ShareLvl****Type**

int

**Default value**

0x1

**Description**

Shareability levels bits in ID\_MMFR0, indicate the number of Shareability levels implemented.

**INITPAHBEN****Type**

bool

**Default value**

0x0

**Description**

The P-AHB enable state at reset.

**INITVTOR\_NS****Type**

int

**Default value**

0x0

**Description**

Non-Secure vector-table offset at reset.

**INITVTOR\_S****Type**

int

**Default value**

0x0

**Description**

Secure vector-table offset at reset.

**IOP****Type**

bool

**Default value**

0x0

**Description**

Send all d-side transactions to the port, io\_port\_out. Transactions which do not match should be returned to the port, io\_port\_in.

**IRQDIS0****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

**IRQDIS1****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

**IRQDIS10****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

**IRQDIS11****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

**IRQDIS12****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

**IRQDIS13****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

**IRQDIS14****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

**IRQDIS15****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+480].

**IRQDIS2****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

**IRQDIS3****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

**IRQDIS4****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

**IRQDIS5****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

**IRQDIS6****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

**IRQDIS7****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

**IRQDIS8****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

**IRQDIS9****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

**ITGU****Type**

bool

**Default value**

0x0

**Description**

ITCM Security Gate Unit included.

**ITGUBLKSZ****Type**

int

**Default value**

0x3

**Description**

ITCM gate unit block size. Size=pow(2, ITGUBLKSZ + 5) bytes.

**ITGUMAXBLKS****Type**

int

**Default value**

0x0

**Description**

Maximum number of ITCM gate unit blocks. Number of blocks=pow(2, ITGUMAXBLKS).

**ITM****Type**

bool

**Default value**

0x1

**Description**

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included (unless baseline).

**ITM\_HAS\_LSR****Type**

bool

**Default value**

0x1

**Description**

ITM support LAR and LSR for software lock.

**LOCKDTGU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Data TGU registers write.

**LOCKITGU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Instruction TGU registers write.

**LOCKTCM****Type**

bool

**Default value**

0x0

**Description**

Lock down of TCM registers write.

**LOCK\_NS\_MPU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Non-Secure MPU registers write.

**LOCK\_SAU****Type**

bool

**Default value**

0x0

**Description**

Lock down of SAU registers write.



**LOCK\_S\_MPU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Secure MPU registers write.

**LVL\_WIDTH****Type**

int

**Default value**

0x3

**Description**

Number of bits of interrupt priority (baseline has 2).

**MEMORY\_REGION\_MASK****Type**

int

**Default value**

0xffffffff

**Description**

Read/Write Mask for MPU\_RBAR, MPU\_RLAR, SAU\_RBAR, SAU\_RLAR. Bits[4:0] of this parameter are ignored.

**MPU\_TYPE\_NS.DREGION****Type**

int

**Default value**

0x10

**Description**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

**MPU\_TYPE\_S.DREGION****Type**

int

**Default value**

0x10

**Description**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

**MVE****Type**

int

**Default value**

0x2

**Description**

Set whether the model has MVE support. If FPU = 0: 0=MVE not included, 1=Integer subset of MVE included. If FPU = 1: 0=MVE not included, 1=Integer subset of MVE included, 2=Integer and half and single precision floating point MVE included.

**MVFR0.Double-precision****Type**

bool

**Default value**

0x1

**Description**

Support 8-byte floats.

**MVFR1.FP16****Type**

bool

**Default value**

0x1

**Description**

FP extension implements half-precision floating-point operations. 0 = Not supported; 1 = Supported.

**MVFR1.FPHP****Type**

int

**Default value**

0x2

**Description**

FP extension implements half-precision floating-point conversion instructions. 0x1: Half-precision to single-precision, 0x2: As for 0x1 and also half-precision to double-precision.

**MVFR1.MVE****Type**

int

**Default value**

0x2

**Description**

DEPRECATED: Use parameter MVE instead.

**NUM\_IRQ****Type**

int

**Default value**

0x10

**Description**

Number of user interrupts.

**NVIC\_ITNS0****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**NVIC\_ITNS1****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**NVIC\_ITNS10****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**NVIC\_ITNS11****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**NVIC\_ITNS12****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**NVIC\_ITNS13****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts

0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

#### **NVIC\_ITNS14**

##### **Type**

string

##### **Default value**

""

##### **Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

#### **NVIC\_ITNS15**

##### **Type**

string

##### **Default value**

""

##### **Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

#### **NVIC\_ITNS2**

##### **Type**

string

##### **Default value**

""

##### **Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

#### **NVIC\_ITNS3**

##### **Type**

string

##### **Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**NVIC\_ITNS4****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**NVIC\_ITNS5****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**NVIC\_ITNS6****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**NVIC\_ITNS7****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**NVIC\_ITNS8****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**NVIC\_ITNS9****Type**

string

**Default value**

""

**Description**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

**REGISTER\_POP\_ORDER****Type**

string

**Default value**

"R4-R11,R0-R3,R12,R14,RETURN\_ADDR,CPSR,S0-S15,FPSCR,PADDING,S16-S31"

**Description**

Order in which the registers are popped off the stack during exception return. A comma separated list of register names and ranges.

**REGISTER\_PUSH\_ORDER****Type**

string

**Default value**

"R0-R3,R12,R14,RETURN\_ADDR,CPSR,S0-S15,FPSCR,PADDING,S16-S31"

**Description**

Order in which the registers are pushed on to the stack during exception handling. A comma separated list of register names and ranges.

**SAU\_CTRL.ALLNS****Type**

bool

**Default value**

0x0

**Description**

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

**SAU\_CTRL.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU at reset.

**SAU\_REGION0.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region0 at reset.

**SAU\_REGION0.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region0 at reset.



**SAU\_REGION0.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region0 at reset.

**SAU\_REGION0.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region0 at reset.

**SAU\_REGION1.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region1 at reset.

**SAU\_REGION1.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region1 at reset.

**SAU\_REGION1.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region1 at reset.

**SAU\_REGION1.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region1 at reset.

**SAU\_REGION10.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region10 at reset.

**SAU\_REGION10.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region10 at reset.

**SAU\_REGION10.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region10 at reset.

**SAU\_REGION10.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region10 at reset.

**SAU\_REGION100.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region100 at reset.

**SAU\_REGION100.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region100 at reset.

**SAU\_REGION100.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region100 at reset.

**SAU\_REGION100.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region100 at reset.

**SAU\_REGION101.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region101 at reset.

**SAU\_REGION101.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region101 at reset.

**SAU\_REGION101.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region101 at reset.

**SAU\_REGION101.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region101 at reset.

**SAU\_REGION102.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region102 at reset.

**SAU\_REGION102.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region102 at reset.

**SAU\_REGION102.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region102 at reset.

**SAU\_REGION102.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region102 at reset.

**SAU\_REGION103.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region103 at reset.

**SAU\_REGION103.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region103 at reset.

**SAU\_REGION103.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region103 at reset.

**SAU\_REGION103.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region103 at reset.

**SAU\_REGION104.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region104 at reset.

**SAU\_REGION104.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region104 at reset.

**SAU\_REGION104.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region104 at reset.

**SAU\_REGION104.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region104 at reset.

**SAU\_REGION105.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region105 at reset.

**SAU\_REGION105.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region105 at reset.

**SAU\_REGION105.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region105 at reset.

**SAU\_REGION105.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region105 at reset.

**SAU\_REGION106.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region106 at reset.

**SAU\_REGION106.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region106 at reset.

**SAU\_REGION106.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region106 at reset.

**SAU\_REGION106.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region106 at reset.

**SAU\_REGION107.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region107 at reset.

**SAU\_REGION107.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region107 at reset.



**SAU\_REGION107.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region107 at reset.

**SAU\_REGION107.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region107 at reset.

**SAU\_REGION108.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region108 at reset.

**SAU\_REGION108.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region108 at reset.

**SAU\_REGION108.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region108 at reset.

**SAU\_REGION108.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region108 at reset.

**SAU\_REGION109.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region109 at reset.

**SAU\_REGION109.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region109 at reset.

**SAU\_REGION109.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region109 at reset.

**SAU\_REGION109.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region109 at reset.

**SAU\_REGION11.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region11 at reset.

**SAU\_REGION11.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region11 at reset.

**SAU\_REGION11.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region11 at reset.

**SAU\_REGION11.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region11 at reset.

**SAU\_REGION110.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region110 at reset.

**SAU\_REGION110.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region110 at reset.

**SAU\_REGION110.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region110 at reset.

**SAU\_REGION110.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region110 at reset.

**SAU\_REGION111.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region111 at reset.

**SAU\_REGION111.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region111 at reset.

**SAU\_REGION111.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region111 at reset.

**SAU\_REGION111.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region111 at reset.

**SAU\_REGION112.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region112 at reset.

**SAU\_REGION112.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region112 at reset.

**SAU\_REGION112.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region112 at reset.

**SAU\_REGION112.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region112 at reset.

**SAU\_REGION113.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region113 at reset.

**SAU\_REGION113.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region113 at reset.

**SAU\_REGION113.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region113 at reset.

**SAU\_REGION113.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region113 at reset.

**SAU\_REGION114.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region114 at reset.

**SAU\_REGION114.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region114 at reset.

**SAU\_REGION114.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region114 at reset.

**SAU\_REGION114.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region114 at reset.

**SAU\_REGION115.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region115 at reset.

**SAU\_REGION115.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region115 at reset.

**SAU\_REGION115.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region115 at reset.

**SAU\_REGION115.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region115 at reset.

**SAU\_REGION116.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region116 at reset.

**SAU\_REGION116.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region116 at reset.



**SAU\_REGION116.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region116 at reset.

**SAU\_REGION116.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region116 at reset.

**SAU\_REGION117.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region117 at reset.

**SAU\_REGION117.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region117 at reset.

**SAU\_REGION117.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region117 at reset.

**SAU\_REGION117.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region117 at reset.

**SAU\_REGION118.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region118 at reset.

**SAU\_REGION118.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region118 at reset.

**SAU\_REGION118.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region118 at reset.

**SAU\_REGION118.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region118 at reset.

**SAU\_REGION119.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region119 at reset.

**SAU\_REGION119.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region119 at reset.

**SAU\_REGION119.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region119 at reset.

**SAU\_REGION119.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region119 at reset.

**SAU\_REGION12.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region12 at reset.

**SAU\_REGION12.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region12 at reset.

**SAU\_REGION12.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region12 at reset.

**SAU\_REGION12.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region12 at reset.

**SAU\_REGION120.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region120 at reset.

**SAU\_REGION120.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region120 at reset.

**SAU\_REGION120.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region120 at reset.

**SAU\_REGION120.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region120 at reset.

**SAU\_REGION121.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region121 at reset.

**SAU\_REGION121.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region121 at reset.

**SAU\_REGION121.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region121 at reset.

**SAU\_REGION121.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region121 at reset.

**SAU\_REGION122.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region122 at reset.

**SAU\_REGION122.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region122 at reset.

**SAU\_REGION122.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region122 at reset.

**SAU\_REGION122.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region122 at reset.

**SAU\_REGION123.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region123 at reset.

**SAU\_REGION123.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region123 at reset.

**SAU\_REGION123.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region123 at reset.

**SAU\_REGION123.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region123 at reset.

**SAU\_REGION124.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region124 at reset.

**SAU\_REGION124.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region124 at reset.

**SAU\_REGION124.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region124 at reset.

**SAU\_REGION124.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region124 at reset.

**SAU\_REGION125.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region125 at reset.

**SAU\_REGION125.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region125 at reset.



**SAU\_REGION125.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region125 at reset.

**SAU\_REGION125.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region125 at reset.

**SAU\_REGION126.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region126 at reset.

**SAU\_REGION126.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region126 at reset.

**SAU\_REGION126.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region126 at reset.

**SAU\_REGION126.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region126 at reset.

**SAU\_REGION127.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region127 at reset.

**SAU\_REGION127.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region127 at reset.

**SAU\_REGION127.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region127 at reset.

**SAU\_REGION127.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region127 at reset.

**SAU\_REGION128.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region128 at reset.

**SAU\_REGION128.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region128 at reset.

**SAU\_REGION128.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region128 at reset.

**SAU\_REGION128.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region128 at reset.

**SAU\_REGION129.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region129 at reset.

**SAU\_REGION129.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region129 at reset.

**SAU\_REGION129.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region129 at reset.

**SAU\_REGION129.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region129 at reset.

**SAU\_REGION13.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region13 at reset.

**SAU\_REGION13.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region13 at reset.

**SAU\_REGION13.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region13 at reset.

**SAU\_REGION13.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region13 at reset.

**SAU\_REGION130.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region130 at reset.

**SAU\_REGION130.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region130 at reset.

**SAU\_REGION130.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region130 at reset.

**SAU\_REGION130.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region130 at reset.

**SAU\_REGION131.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region131 at reset.

**SAU\_REGION131.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region131 at reset.

**SAU\_REGION131.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region131 at reset.

**SAU\_REGION131.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region131 at reset.

**SAU\_REGION132.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region132 at reset.

**SAU\_REGION132.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region132 at reset.

**SAU\_REGION132.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region132 at reset.

**SAU\_REGION132.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region132 at reset.

**SAU\_REGION133.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region133 at reset.

**SAU\_REGION133.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region133 at reset.

**SAU\_REGION133.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region133 at reset.

**SAU\_REGION133.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region133 at reset.

**SAU\_REGION134.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region134 at reset.

**SAU\_REGION134.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region134 at reset.



**SAU\_REGION134.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region134 at reset.

**SAU\_REGION134.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region134 at reset.

**SAU\_REGION135.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region135 at reset.

**SAU\_REGION135.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region135 at reset.

**SAU\_REGION135.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region135 at reset.

**SAU\_REGION135.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region135 at reset.

**SAU\_REGION136.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region136 at reset.

**SAU\_REGION136.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region136 at reset.

**SAU\_REGION136.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region136 at reset.

**SAU\_REGION136.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region136 at reset.

**SAU\_REGION137.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region137 at reset.

**SAU\_REGION137.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region137 at reset.

**SAU\_REGION137.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region137 at reset.

**SAU\_REGION137.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region137 at reset.

**SAU\_REGION138.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region138 at reset.

**SAU\_REGION138.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region138 at reset.

**SAU\_REGION138.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region138 at reset.

**SAU\_REGION138.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region138 at reset.

**SAU\_REGION139.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region139 at reset.

**SAU\_REGION139.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region139 at reset.

**SAU\_REGION139.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region139 at reset.

**SAU\_REGION139.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region139 at reset.

**SAU\_REGION14.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region14 at reset.

**SAU\_REGION14.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region14 at reset.

**SAU\_REGION14.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region14 at reset.

**SAU\_REGION14.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region14 at reset.

**SAU\_REGION140.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region140 at reset.

**SAU\_REGION140.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region140 at reset.

**SAU\_REGION140.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region140 at reset.

**SAU\_REGION140.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region140 at reset.

**SAU\_REGION141.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region141 at reset.

**SAU\_REGION141.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region141 at reset.

**SAU\_REGION141.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region141 at reset.

**SAU\_REGION141.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region141 at reset.

**SAU\_REGION142.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region142 at reset.

**SAU\_REGION142.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region142 at reset.

**SAU\_REGION142.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region142 at reset.

**SAU\_REGION142.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region142 at reset.

**SAU\_REGION143.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region143 at reset.

**SAU\_REGION143.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region143 at reset.



**SAU\_REGION143.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region143 at reset.

**SAU\_REGION143.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region143 at reset.

**SAU\_REGION144.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region144 at reset.

**SAU\_REGION144.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region144 at reset.

**SAU\_REGION144.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region144 at reset.

**SAU\_REGION144.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region144 at reset.

**SAU\_REGION145.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region145 at reset.

**SAU\_REGION145.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region145 at reset.

**SAU\_REGION145.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region145 at reset.

**SAU\_REGION145.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region145 at reset.

**SAU\_REGION146.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region146 at reset.

**SAU\_REGION146.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region146 at reset.

**SAU\_REGION146.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region146 at reset.

**SAU\_REGION146.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region146 at reset.

**SAU\_REGION147.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region147 at reset.

**SAU\_REGION147.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region147 at reset.

**SAU\_REGION147.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region147 at reset.

**SAU\_REGION147.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region147 at reset.

**SAU\_REGION148.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region148 at reset.

**SAU\_REGION148.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region148 at reset.

**SAU\_REGION148.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region148 at reset.

**SAU\_REGION148.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region148 at reset.

**SAU\_REGION149.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region149 at reset.

**SAU\_REGION149.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region149 at reset.

**SAU\_REGION149.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region149 at reset.

**SAU\_REGION149.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region149 at reset.

**SAU\_REGION15.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region15 at reset.

**SAU\_REGION15.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region15 at reset.

**SAU\_REGION15.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region15 at reset.

**SAU\_REGION15.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region15 at reset.

**SAU\_REGION150.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region150 at reset.

**SAU\_REGION150.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region150 at reset.

**SAU\_REGION150.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region150 at reset.

**SAU\_REGION150.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region150 at reset.

**SAU\_REGION151.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region151 at reset.

**SAU\_REGION151.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region151 at reset.

**SAU\_REGION151.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region151 at reset.

**SAU\_REGION151.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region151 at reset.

**SAU\_REGION152.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region152 at reset.

**SAU\_REGION152.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region152 at reset.



**SAU\_REGION152.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region152 at reset.

**SAU\_REGION152.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region152 at reset.

**SAU\_REGION153.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region153 at reset.

**SAU\_REGION153.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region153 at reset.

**SAU\_REGION153.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region153 at reset.

**SAU\_REGION153.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region153 at reset.

**SAU\_REGION154.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region154 at reset.

**SAU\_REGION154.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region154 at reset.

**SAU\_REGION154.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region154 at reset.

**SAU\_REGION154.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region154 at reset.

**SAU\_REGION155.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region155 at reset.

**SAU\_REGION155.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region155 at reset.

**SAU\_REGION155.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region155 at reset.

**SAU\_REGION155.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region155 at reset.

**SAU\_REGION156.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region156 at reset.

**SAU\_REGION156.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region156 at reset.

**SAU\_REGION156.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region156 at reset.

**SAU\_REGION156.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region156 at reset.

**SAU\_REGION157.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region157 at reset.

**SAU\_REGION157.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region157 at reset.

**SAU\_REGION157.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region157 at reset.

**SAU\_REGION157.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region157 at reset.

**SAU\_REGION158.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region158 at reset.

**SAU\_REGION158.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region158 at reset.

**SAU\_REGION158.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region158 at reset.

**SAU\_REGION158.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region158 at reset.

**SAU\_REGION159.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region159 at reset.

**SAU\_REGION159.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region159 at reset.

**SAU\_REGION159.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region159 at reset.

**SAU\_REGION159.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region159 at reset.

**SAU\_REGION16.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region16 at reset.

**SAU\_REGION16.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region16 at reset.

**SAU\_REGION16.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region16 at reset.

**SAU\_REGION16.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region16 at reset.

**SAU\_REGION160.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region160 at reset.

**SAU\_REGION160.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region160 at reset.

**SAU\_REGION160.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region160 at reset.

**SAU\_REGION160.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region160 at reset.

**SAU\_REGION161.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region161 at reset.

**SAU\_REGION161.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region161 at reset.



**SAU\_REGION161.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region161 at reset.

**SAU\_REGION161.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region161 at reset.

**SAU\_REGION162.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region162 at reset.

**SAU\_REGION162.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region162 at reset.

**SAU\_REGION162.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region162 at reset.

**SAU\_REGION162.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region162 at reset.

**SAU\_REGION163.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region163 at reset.

**SAU\_REGION163.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region163 at reset.

**SAU\_REGION163.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region163 at reset.

**SAU\_REGION163.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region163 at reset.

**SAU\_REGION164.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region164 at reset.

**SAU\_REGION164.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region164 at reset.

**SAU\_REGION164.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region164 at reset.

**SAU\_REGION164.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region164 at reset.

**SAU\_REGION165.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region165 at reset.

**SAU\_REGION165.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region165 at reset.

**SAU\_REGION165.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region165 at reset.

**SAU\_REGION165.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region165 at reset.

**SAU\_REGION166.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region166 at reset.

**SAU\_REGION166.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region166 at reset.

**SAU\_REGION166.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region166 at reset.

**SAU\_REGION166.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region166 at reset.

**SAU\_REGION167.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region167 at reset.

**SAU\_REGION167.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region167 at reset.

**SAU\_REGION167.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region167 at reset.

**SAU\_REGION167.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region167 at reset.

**SAU\_REGION168.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region168 at reset.

**SAU\_REGION168.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region168 at reset.

**SAU\_REGION168.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region168 at reset.

**SAU\_REGION168.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region168 at reset.

**SAU\_REGION169.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region169 at reset.

**SAU\_REGION169.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region169 at reset.

**SAU\_REGION169.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region169 at reset.

**SAU\_REGION169.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region169 at reset.

**SAU\_REGION17.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region17 at reset.

**SAU\_REGION17.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region17 at reset.

**SAU\_REGION17.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region17 at reset.

**SAU\_REGION17.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region17 at reset.

**SAU\_REGION170.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region170 at reset.

**SAU\_REGION170.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region170 at reset.



**SAU\_REGION170.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region170 at reset.

**SAU\_REGION170.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region170 at reset.

**SAU\_REGION171.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region171 at reset.

**SAU\_REGION171.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region171 at reset.

**SAU\_REGION171.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region171 at reset.

**SAU\_REGION171.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region171 at reset.

**SAU\_REGION172.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region172 at reset.

**SAU\_REGION172.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region172 at reset.

**SAU\_REGION172.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region172 at reset.

**SAU\_REGION172.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region172 at reset.

**SAU\_REGION173.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region173 at reset.

**SAU\_REGION173.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region173 at reset.

**SAU\_REGION173.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region173 at reset.

**SAU\_REGION173.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region173 at reset.

**SAU\_REGION174.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region174 at reset.

**SAU\_REGION174.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region174 at reset.

**SAU\_REGION174.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region174 at reset.

**SAU\_REGION174.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region174 at reset.

**SAU\_REGION175.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region175 at reset.

**SAU\_REGION175.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region175 at reset.

**SAU\_REGION175.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region175 at reset.

**SAU\_REGION175.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region175 at reset.

**SAU\_REGION176.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region176 at reset.

**SAU\_REGION176.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region176 at reset.

**SAU\_REGION176.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region176 at reset.

**SAU\_REGION176.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region176 at reset.

**SAU\_REGION177.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region177 at reset.

**SAU\_REGION177.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region177 at reset.

**SAU\_REGION177.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region177 at reset.

**SAU\_REGION177.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region177 at reset.

**SAU\_REGION178.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region178 at reset.

**SAU\_REGION178.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region178 at reset.

**SAU\_REGION178.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region178 at reset.

**SAU\_REGION178.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region178 at reset.

**SAU\_REGION179.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region179 at reset.

**SAU\_REGION179.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region179 at reset.

**SAU\_REGION179.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region179 at reset.

**SAU\_REGION179.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region179 at reset.

**SAU\_REGION18.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region18 at reset.

**SAU\_REGION18.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region18 at reset.



**SAU\_REGION18.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region18 at reset.

**SAU\_REGION18.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region18 at reset.

**SAU\_REGION180.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region180 at reset.

**SAU\_REGION180.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region180 at reset.

**SAU\_REGION180.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region180 at reset.

**SAU\_REGION180.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region180 at reset.

**SAU\_REGION181.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region181 at reset.

**SAU\_REGION181.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region181 at reset.

**SAU\_REGION181.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region181 at reset.

**SAU\_REGION181.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region181 at reset.

**SAU\_REGION182.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region182 at reset.

**SAU\_REGION182.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region182 at reset.

**SAU\_REGION182.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region182 at reset.

**SAU\_REGION182.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region182 at reset.

**SAU\_REGION183.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region183 at reset.

**SAU\_REGION183.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region183 at reset.

**SAU\_REGION183.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region183 at reset.

**SAU\_REGION183.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region183 at reset.

**SAU\_REGION184.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region184 at reset.

**SAU\_REGION184.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region184 at reset.

**SAU\_REGION184.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region184 at reset.

**SAU\_REGION184.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region184 at reset.

**SAU\_REGION185.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region185 at reset.

**SAU\_REGION185.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region185 at reset.

**SAU\_REGION185.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region185 at reset.

**SAU\_REGION185.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region185 at reset.

**SAU\_REGION186.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region186 at reset.

**SAU\_REGION186.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region186 at reset.

**SAU\_REGION186.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region186 at reset.

**SAU\_REGION186.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region186 at reset.

**SAU\_REGION187.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region187 at reset.

**SAU\_REGION187.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region187 at reset.

**SAU\_REGION187.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region187 at reset.

**SAU\_REGION187.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region187 at reset.

**SAU\_REGION188.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region188 at reset.

**SAU\_REGION188.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region188 at reset.

**SAU\_REGION188.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region188 at reset.

**SAU\_REGION188.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region188 at reset.

**SAU\_REGION189.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region189 at reset.

**SAU\_REGION189.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region189 at reset.



**SAU\_REGION189.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region189 at reset.

**SAU\_REGION189.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region189 at reset.

**SAU\_REGION19.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region19 at reset.

**SAU\_REGION19.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region19 at reset.

**SAU\_REGION19.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region19 at reset.

**SAU\_REGION19.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region19 at reset.

**SAU\_REGION190.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region190 at reset.

**SAU\_REGION190.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region190 at reset.

**SAU\_REGION190.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region190 at reset.

**SAU\_REGION190.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region190 at reset.

**SAU\_REGION191.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region191 at reset.

**SAU\_REGION191.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region191 at reset.

**SAU\_REGION191.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region191 at reset.

**SAU\_REGION191.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region191 at reset.

**SAU\_REGION192.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region192 at reset.

**SAU\_REGION192.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region192 at reset.

**SAU\_REGION192.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region192 at reset.

**SAU\_REGION192.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region192 at reset.

**SAU\_REGION193.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region193 at reset.

**SAU\_REGION193.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region193 at reset.

**SAU\_REGION193.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region193 at reset.

**SAU\_REGION193.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region193 at reset.

**SAU\_REGION194.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region194 at reset.

**SAU\_REGION194.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region194 at reset.

**SAU\_REGION194.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region194 at reset.

**SAU\_REGION194.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region194 at reset.

**SAU\_REGION195.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region195 at reset.

**SAU\_REGION195.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region195 at reset.

**SAU\_REGION195.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region195 at reset.

**SAU\_REGION195.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region195 at reset.

**SAU\_REGION196.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region196 at reset.

**SAU\_REGION196.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region196 at reset.

**SAU\_REGION196.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region196 at reset.

**SAU\_REGION196.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region196 at reset.

**SAU\_REGION197.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region197 at reset.

**SAU\_REGION197.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region197 at reset.

**SAU\_REGION197.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region197 at reset.

**SAU\_REGION197.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region197 at reset.

**SAU\_REGION198.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region198 at reset.

**SAU\_REGION198.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region198 at reset.



**SAU\_REGION198.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region198 at reset.

**SAU\_REGION198.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region198 at reset.

**SAU\_REGION199.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region199 at reset.

**SAU\_REGION199.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region199 at reset.

**SAU\_REGION199.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region199 at reset.

**SAU\_REGION199.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region199 at reset.

**SAU\_REGION2.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region2 at reset.

**SAU\_REGION2.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region2 at reset.

**SAU\_REGION2.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region2 at reset.

**SAU\_REGION2.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region2 at reset.

**SAU\_REGION20.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region20 at reset.

**SAU\_REGION20.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region20 at reset.

**SAU\_REGION20.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region20 at reset.

**SAU\_REGION20.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region20 at reset.

**SAU\_REGION200.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region200 at reset.

**SAU\_REGION200.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region200 at reset.

**SAU\_REGION200.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region200 at reset.

**SAU\_REGION200.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region200 at reset.

**SAU\_REGION201.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region201 at reset.

**SAU\_REGION201.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region201 at reset.

**SAU\_REGION201.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region201 at reset.

**SAU\_REGION201.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region201 at reset.

**SAU\_REGION202.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region202 at reset.

**SAU\_REGION202.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region202 at reset.

**SAU\_REGION202.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region202 at reset.

**SAU\_REGION202.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region202 at reset.

**SAU\_REGION203.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region203 at reset.

**SAU\_REGION203.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region203 at reset.

**SAU\_REGION203.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region203 at reset.

**SAU\_REGION203.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region203 at reset.

**SAU\_REGION204.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region204 at reset.

**SAU\_REGION204.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region204 at reset.

**SAU\_REGION204.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region204 at reset.

**SAU\_REGION204.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region204 at reset.

**SAU\_REGION205.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region205 at reset.

**SAU\_REGION205.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region205 at reset.

**SAU\_REGION205.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region205 at reset.

**SAU\_REGION205.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region205 at reset.

**SAU\_REGION206.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region206 at reset.

**SAU\_REGION206.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region206 at reset.



**SAU\_REGION206.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region206 at reset.

**SAU\_REGION206.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region206 at reset.

**SAU\_REGION207.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region207 at reset.

**SAU\_REGION207.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region207 at reset.

**SAU\_REGION207.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region207 at reset.

**SAU\_REGION207.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region207 at reset.

**SAU\_REGION208.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region208 at reset.

**SAU\_REGION208.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region208 at reset.

**SAU\_REGION208.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region208 at reset.

**SAU\_REGION208.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region208 at reset.

**SAU\_REGION209.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region209 at reset.

**SAU\_REGION209.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region209 at reset.

**SAU\_REGION209.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region209 at reset.

**SAU\_REGION209.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region209 at reset.

**SAU\_REGION21.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region21 at reset.

**SAU\_REGION21.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region21 at reset.

**SAU\_REGION21.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region21 at reset.

**SAU\_REGION21.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region21 at reset.

**SAU\_REGION210.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region210 at reset.

**SAU\_REGION210.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region210 at reset.

**SAU\_REGION210.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region210 at reset.

**SAU\_REGION210.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region210 at reset.

**SAU\_REGION211.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region211 at reset.

**SAU\_REGION211.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region211 at reset.

**SAU\_REGION211.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region211 at reset.

**SAU\_REGION211.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region211 at reset.

**SAU\_REGION212.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region212 at reset.

**SAU\_REGION212.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region212 at reset.

**SAU\_REGION212.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region212 at reset.

**SAU\_REGION212.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region212 at reset.

**SAU\_REGION213.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region213 at reset.

**SAU\_REGION213.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region213 at reset.

**SAU\_REGION213.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region213 at reset.

**SAU\_REGION213.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region213 at reset.

**SAU\_REGION214.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region214 at reset.

**SAU\_REGION214.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region214 at reset.

**SAU\_REGION214.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region214 at reset.

**SAU\_REGION214.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region214 at reset.

**SAU\_REGION215.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region215 at reset.

**SAU\_REGION215.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region215 at reset.



**SAU\_REGION215.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region215 at reset.

**SAU\_REGION215.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region215 at reset.

**SAU\_REGION216.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region216 at reset.

**SAU\_REGION216.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region216 at reset.

**SAU\_REGION216.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region216 at reset.

**SAU\_REGION216.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region216 at reset.

**SAU\_REGION217.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region217 at reset.

**SAU\_REGION217.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region217 at reset.

**SAU\_REGION217.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region217 at reset.

**SAU\_REGION217.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region217 at reset.

**SAU\_REGION218.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region218 at reset.

**SAU\_REGION218.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region218 at reset.

**SAU\_REGION218.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region218 at reset.

**SAU\_REGION218.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region218 at reset.

**SAU\_REGION219.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region219 at reset.

**SAU\_REGION219.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region219 at reset.

**SAU\_REGION219.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region219 at reset.

**SAU\_REGION219.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region219 at reset.

**SAU\_REGION22.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region22 at reset.

**SAU\_REGION22.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region22 at reset.

**SAU\_REGION22.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region22 at reset.

**SAU\_REGION22.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region22 at reset.

**SAU\_REGION220.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region220 at reset.

**SAU\_REGION220.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region220 at reset.

**SAU\_REGION220.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region220 at reset.

**SAU\_REGION220.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region220 at reset.

**SAU\_REGION221.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region221 at reset.

**SAU\_REGION221.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region221 at reset.

**SAU\_REGION221.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region221 at reset.

**SAU\_REGION221.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region221 at reset.

**SAU\_REGION222.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region222 at reset.

**SAU\_REGION222.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region222 at reset.

**SAU\_REGION222.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region222 at reset.

**SAU\_REGION222.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region222 at reset.

**SAU\_REGION223.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region223 at reset.

**SAU\_REGION223.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region223 at reset.

**SAU\_REGION223.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region223 at reset.

**SAU\_REGION223.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region223 at reset.

**SAU\_REGION224.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region224 at reset.

**SAU\_REGION224.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region224 at reset.



**SAU\_REGION224.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region224 at reset.

**SAU\_REGION224.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region224 at reset.

**SAU\_REGION225.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region225 at reset.

**SAU\_REGION225.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region225 at reset.

**SAU\_REGION225.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region225 at reset.

**SAU\_REGION225.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region225 at reset.

**SAU\_REGION226.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region226 at reset.

**SAU\_REGION226.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region226 at reset.

**SAU\_REGION226.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region226 at reset.

**SAU\_REGION226.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region226 at reset.

**SAU\_REGION227.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region227 at reset.

**SAU\_REGION227.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region227 at reset.

**SAU\_REGION227.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region227 at reset.

**SAU\_REGION227.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region227 at reset.

**SAU\_REGION228.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region228 at reset.

**SAU\_REGION228.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region228 at reset.

**SAU\_REGION228.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region228 at reset.

**SAU\_REGION228.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region228 at reset.

**SAU\_REGION229.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region229 at reset.

**SAU\_REGION229.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region229 at reset.

**SAU\_REGION229.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region229 at reset.

**SAU\_REGION229.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region229 at reset.

**SAU\_REGION23.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region23 at reset.

**SAU\_REGION23.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region23 at reset.

**SAU\_REGION23.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region23 at reset.

**SAU\_REGION23.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region23 at reset.

**SAU\_REGION230.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region230 at reset.

**SAU\_REGION230.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region230 at reset.

**SAU\_REGION230.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region230 at reset.

**SAU\_REGION230.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region230 at reset.

**SAU\_REGION231.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region231 at reset.

**SAU\_REGION231.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region231 at reset.

**SAU\_REGION231.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region231 at reset.

**SAU\_REGION231.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region231 at reset.

**SAU\_REGION232.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region232 at reset.

**SAU\_REGION232.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region232 at reset.

**SAU\_REGION232.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region232 at reset.

**SAU\_REGION232.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region232 at reset.

**SAU\_REGION233.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region233 at reset.

**SAU\_REGION233.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region233 at reset.



**SAU\_REGION233.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region233 at reset.

**SAU\_REGION233.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region233 at reset.

**SAU\_REGION234.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region234 at reset.

**SAU\_REGION234.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region234 at reset.

**SAU\_REGION234.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region234 at reset.

**SAU\_REGION234.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region234 at reset.

**SAU\_REGION235.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region235 at reset.

**SAU\_REGION235.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region235 at reset.

**SAU\_REGION235.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region235 at reset.

**SAU\_REGION235.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region235 at reset.

**SAU\_REGION236.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region236 at reset.

**SAU\_REGION236.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region236 at reset.

**SAU\_REGION236.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region236 at reset.

**SAU\_REGION236.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region236 at reset.

**SAU\_REGION237.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region237 at reset.

**SAU\_REGION237.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region237 at reset.

**SAU\_REGION237.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region237 at reset.

**SAU\_REGION237.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region237 at reset.

**SAU\_REGION238.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region238 at reset.

**SAU\_REGION238.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region238 at reset.

**SAU\_REGION238.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region238 at reset.

**SAU\_REGION238.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region238 at reset.

**SAU\_REGION239.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region239 at reset.

**SAU\_REGION239.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region239 at reset.

**SAU\_REGION239.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region239 at reset.

**SAU\_REGION239.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region239 at reset.

**SAU\_REGION24.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region24 at reset.

**SAU\_REGION24.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region24 at reset.

**SAU\_REGION24.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region24 at reset.

**SAU\_REGION24.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region24 at reset.

**SAU\_REGION240.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region240 at reset.

**SAU\_REGION240.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region240 at reset.

**SAU\_REGION240.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region240 at reset.

**SAU\_REGION240.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region240 at reset.

**SAU\_REGION241.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region241 at reset.

**SAU\_REGION241.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region241 at reset.

**SAU\_REGION241.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region241 at reset.

**SAU\_REGION241.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region241 at reset.

**SAU\_REGION242.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region242 at reset.

**SAU\_REGION242.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region242 at reset.



**SAU\_REGION242.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region242 at reset.

**SAU\_REGION242.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region242 at reset.

**SAU\_REGION243.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region243 at reset.

**SAU\_REGION243.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region243 at reset.

**SAU\_REGION243.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region243 at reset.

**SAU\_REGION243.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region243 at reset.

**SAU\_REGION244.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region244 at reset.

**SAU\_REGION244.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region244 at reset.

**SAU\_REGION244.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region244 at reset.

**SAU\_REGION244.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region244 at reset.

**SAU\_REGION245.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region245 at reset.

**SAU\_REGION245.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region245 at reset.

**SAU\_REGION245.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region245 at reset.

**SAU\_REGION245.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region245 at reset.

**SAU\_REGION246.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region246 at reset.

**SAU\_REGION246.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region246 at reset.

**SAU\_REGION246.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region246 at reset.

**SAU\_REGION246.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region246 at reset.

**SAU\_REGION247.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region247 at reset.

**SAU\_REGION247.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region247 at reset.

**SAU\_REGION247.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region247 at reset.

**SAU\_REGION247.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region247 at reset.

**SAU\_REGION248.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region248 at reset.

**SAU\_REGION248.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region248 at reset.

**SAU\_REGION248.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region248 at reset.

**SAU\_REGION248.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region248 at reset.

**SAU\_REGION249.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region249 at reset.

**SAU\_REGION249.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region249 at reset.

**SAU\_REGION249.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region249 at reset.

**SAU\_REGION249.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region249 at reset.

**SAU\_REGION25.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region25 at reset.

**SAU\_REGION25.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region25 at reset.

**SAU\_REGION25.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region25 at reset.

**SAU\_REGION25.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region25 at reset.

**SAU\_REGION250.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region250 at reset.

**SAU\_REGION250.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region250 at reset.

**SAU\_REGION250.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region250 at reset.

**SAU\_REGION250.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region250 at reset.

**SAU\_REGION251.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region251 at reset.

**SAU\_REGION251.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region251 at reset.



**SAU\_REGION251.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region251 at reset.

**SAU\_REGION251.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region251 at reset.

**SAU\_REGION252.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region252 at reset.

**SAU\_REGION252.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region252 at reset.

**SAU\_REGION252.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region252 at reset.

**SAU\_REGION252.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region252 at reset.

**SAU\_REGION253.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region253 at reset.

**SAU\_REGION253.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region253 at reset.

**SAU\_REGION253.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region253 at reset.

**SAU\_REGION253.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region253 at reset.

**SAU\_REGION254.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region254 at reset.

**SAU\_REGION254.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region254 at reset.

**SAU\_REGION254.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region254 at reset.

**SAU\_REGION254.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region254 at reset.

**SAU\_REGION255.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region255 at reset.

**SAU\_REGION255.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region255 at reset.

**SAU\_REGION255.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region255 at reset.

**SAU\_REGION255.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region255 at reset.

**SAU\_REGION26.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region26 at reset.

**SAU\_REGION26.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region26 at reset.

**SAU\_REGION26.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region26 at reset.

**SAU\_REGION26.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region26 at reset.

**SAU\_REGION27.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region27 at reset.

**SAU\_REGION27.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region27 at reset.

**SAU\_REGION27.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region27 at reset.

**SAU\_REGION27.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region27 at reset.

**SAU\_REGION28.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region28 at reset.

**SAU\_REGION28.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region28 at reset.

**SAU\_REGION28.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region28 at reset.

**SAU\_REGION28.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region28 at reset.

**SAU\_REGION29.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region29 at reset.

**SAU\_REGION29.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region29 at reset.

**SAU\_REGION29.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region29 at reset.

**SAU\_REGION29.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region29 at reset.

**SAU\_REGION3.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region3 at reset.

**SAU\_REGION3.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region3 at reset.

**SAU\_REGION3.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region3 at reset.

**SAU\_REGION3.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region3 at reset.

**SAU\_REGION30.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region30 at reset.

**SAU\_REGION30.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region30 at reset.



**SAU\_REGION30.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region30 at reset.

**SAU\_REGION30.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region30 at reset.

**SAU\_REGION31.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region31 at reset.

**SAU\_REGION31.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region31 at reset.

**SAU\_REGION31.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region31 at reset.

**SAU\_REGION31.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region31 at reset.

**SAU\_REGION32.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region32 at reset.

**SAU\_REGION32.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region32 at reset.

**SAU\_REGION32.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region32 at reset.

**SAU\_REGION32.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region32 at reset.

**SAU\_REGION33.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region33 at reset.

**SAU\_REGION33.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region33 at reset.

**SAU\_REGION33.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region33 at reset.

**SAU\_REGION33.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region33 at reset.

**SAU\_REGION34.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region34 at reset.

**SAU\_REGION34.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region34 at reset.

**SAU\_REGION34.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region34 at reset.

**SAU\_REGION34.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region34 at reset.

**SAU\_REGION35.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region35 at reset.

**SAU\_REGION35.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region35 at reset.

**SAU\_REGION35.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region35 at reset.

**SAU\_REGION35.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region35 at reset.

**SAU\_REGION36.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region36 at reset.

**SAU\_REGION36.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region36 at reset.

**SAU\_REGION36.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region36 at reset.

**SAU\_REGION36.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region36 at reset.

**SAU\_REGION37.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region37 at reset.

**SAU\_REGION37.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region37 at reset.

**SAU\_REGION37.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region37 at reset.

**SAU\_REGION37.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region37 at reset.

**SAU\_REGION38.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region38 at reset.

**SAU\_REGION38.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region38 at reset.

**SAU\_REGION38.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region38 at reset.

**SAU\_REGION38.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region38 at reset.

**SAU\_REGION39.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region39 at reset.

**SAU\_REGION39.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region39 at reset.

**SAU\_REGION39.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region39 at reset.

**SAU\_REGION39.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region39 at reset.

**SAU\_REGION4.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region4 at reset.

**SAU\_REGION4.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region4 at reset.



**SAU\_REGION4.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region4 at reset.

**SAU\_REGION4.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region4 at reset.

**SAU\_REGION40.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region40 at reset.

**SAU\_REGION40.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region40 at reset.

**SAU\_REGION40.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region40 at reset.

**SAU\_REGION40.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region40 at reset.

**SAU\_REGION41.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region41 at reset.

**SAU\_REGION41.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region41 at reset.

**SAU\_REGION41.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region41 at reset.

**SAU\_REGION41.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region41 at reset.

**SAU\_REGION42.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region42 at reset.

**SAU\_REGION42.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region42 at reset.

**SAU\_REGION42.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region42 at reset.

**SAU\_REGION42.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region42 at reset.

**SAU\_REGION43.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region43 at reset.

**SAU\_REGION43.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region43 at reset.

**SAU\_REGION43.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region43 at reset.

**SAU\_REGION43.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region43 at reset.

**SAU\_REGION44.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region44 at reset.

**SAU\_REGION44.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region44 at reset.

**SAU\_REGION44.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region44 at reset.

**SAU\_REGION44.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region44 at reset.

**SAU\_REGION45.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region45 at reset.

**SAU\_REGION45.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region45 at reset.

**SAU\_REGION45.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region45 at reset.

**SAU\_REGION45.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region45 at reset.

**SAU\_REGION46.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region46 at reset.

**SAU\_REGION46.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region46 at reset.

**SAU\_REGION46.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region46 at reset.

**SAU\_REGION46.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region46 at reset.

**SAU\_REGION47.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region47 at reset.

**SAU\_REGION47.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region47 at reset.

**SAU\_REGION47.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region47 at reset.

**SAU\_REGION47.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region47 at reset.

**SAU\_REGION48.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region48 at reset.

**SAU\_REGION48.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region48 at reset.

**SAU\_REGION48.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region48 at reset.

**SAU\_REGION48.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region48 at reset.

**SAU\_REGION49.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region49 at reset.

**SAU\_REGION49.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region49 at reset.



**SAU\_REGION49.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region49 at reset.

**SAU\_REGION49.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region49 at reset.

**SAU\_REGION5.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region5 at reset.

**SAU\_REGION5.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region5 at reset.

**SAU\_REGION5.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region5 at reset.

**SAU\_REGION5.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region5 at reset.

**SAU\_REGION50.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region50 at reset.

**SAU\_REGION50.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region50 at reset.

**SAU\_REGION50.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region50 at reset.

**SAU\_REGION50.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region50 at reset.

**SAU\_REGION51.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region51 at reset.

**SAU\_REGION51.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region51 at reset.

**SAU\_REGION51.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region51 at reset.

**SAU\_REGION51.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region51 at reset.

**SAU\_REGION52.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region52 at reset.

**SAU\_REGION52.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region52 at reset.

**SAU\_REGION52.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region52 at reset.

**SAU\_REGION52.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region52 at reset.

**SAU\_REGION53.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region53 at reset.

**SAU\_REGION53.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region53 at reset.

**SAU\_REGION53.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region53 at reset.

**SAU\_REGION53.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region53 at reset.

**SAU\_REGION54.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region54 at reset.

**SAU\_REGION54.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region54 at reset.

**SAU\_REGION54.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region54 at reset.

**SAU\_REGION54.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region54 at reset.

**SAU\_REGION55.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region55 at reset.

**SAU\_REGION55.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region55 at reset.

**SAU\_REGION55.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region55 at reset.

**SAU\_REGION55.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region55 at reset.

**SAU\_REGION56.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region56 at reset.

**SAU\_REGION56.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region56 at reset.

**SAU\_REGION56.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region56 at reset.

**SAU\_REGION56.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region56 at reset.

**SAU\_REGION57.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region57 at reset.

**SAU\_REGION57.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region57 at reset.

**SAU\_REGION57.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region57 at reset.

**SAU\_REGION57.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region57 at reset.

**SAU\_REGION58.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region58 at reset.

**SAU\_REGION58.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region58 at reset.



**SAU\_REGION58.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region58 at reset.

**SAU\_REGION58.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region58 at reset.

**SAU\_REGION59.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region59 at reset.

**SAU\_REGION59.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region59 at reset.

**SAU\_REGION59.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region59 at reset.

**SAU\_REGION59.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region59 at reset.

**SAU\_REGION6.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region6 at reset.

**SAU\_REGION6.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region6 at reset.

**SAU\_REGION6.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region6 at reset.

**SAU\_REGION6.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region6 at reset.

**SAU\_REGION60.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region60 at reset.

**SAU\_REGION60.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region60 at reset.

**SAU\_REGION60.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region60 at reset.

**SAU\_REGION60.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region60 at reset.

**SAU\_REGION61.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region61 at reset.

**SAU\_REGION61.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region61 at reset.

**SAU\_REGION61.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region61 at reset.

**SAU\_REGION61.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region61 at reset.

**SAU\_REGION62.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region62 at reset.

**SAU\_REGION62.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region62 at reset.

**SAU\_REGION62.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region62 at reset.

**SAU\_REGION62.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region62 at reset.

**SAU\_REGION63.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region63 at reset.

**SAU\_REGION63.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region63 at reset.

**SAU\_REGION63.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region63 at reset.

**SAU\_REGION63.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region63 at reset.

**SAU\_REGION64.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region64 at reset.

**SAU\_REGION64.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region64 at reset.

**SAU\_REGION64.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region64 at reset.

**SAU\_REGION64.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region64 at reset.

**SAU\_REGION65.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region65 at reset.

**SAU\_REGION65.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region65 at reset.

**SAU\_REGION65.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region65 at reset.

**SAU\_REGION65.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region65 at reset.

**SAU\_REGION66.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region66 at reset.

**SAU\_REGION66.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region66 at reset.

**SAU\_REGION66.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region66 at reset.

**SAU\_REGION66.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region66 at reset.

**SAU\_REGION67.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region67 at reset.

**SAU\_REGION67.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region67 at reset.



**SAU\_REGION67.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region67 at reset.

**SAU\_REGION67.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region67 at reset.

**SAU\_REGION68.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region68 at reset.

**SAU\_REGION68.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region68 at reset.

**SAU\_REGION68.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region68 at reset.

**SAU\_REGION68.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region68 at reset.

**SAU\_REGION69.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region69 at reset.

**SAU\_REGION69.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region69 at reset.

**SAU\_REGION69.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region69 at reset.

**SAU\_REGION69.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region69 at reset.

**SAU\_REGION7.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region7 at reset.

**SAU\_REGION7.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region7 at reset.

**SAU\_REGION7.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region7 at reset.

**SAU\_REGION7.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region7 at reset.

**SAU\_REGION70.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region70 at reset.

**SAU\_REGION70.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region70 at reset.

**SAU\_REGION70.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region70 at reset.

**SAU\_REGION70.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region70 at reset.

**SAU\_REGION71.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region71 at reset.

**SAU\_REGION71.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region71 at reset.

**SAU\_REGION71.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region71 at reset.

**SAU\_REGION71.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region71 at reset.

**SAU\_REGION72.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region72 at reset.

**SAU\_REGION72.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region72 at reset.

**SAU\_REGION72.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region72 at reset.

**SAU\_REGION72.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region72 at reset.

**SAU\_REGION73.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region73 at reset.

**SAU\_REGION73.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region73 at reset.

**SAU\_REGION73.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region73 at reset.

**SAU\_REGION73.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region73 at reset.

**SAU\_REGION74.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region74 at reset.

**SAU\_REGION74.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region74 at reset.

**SAU\_REGION74.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region74 at reset.

**SAU\_REGION74.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region74 at reset.

**SAU\_REGION75.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region75 at reset.

**SAU\_REGION75.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region75 at reset.

**SAU\_REGION75.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region75 at reset.

**SAU\_REGION75.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region75 at reset.

**SAU\_REGION76.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region76 at reset.

**SAU\_REGION76.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region76 at reset.



**SAU\_REGION76.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region76 at reset.

**SAU\_REGION76.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region76 at reset.

**SAU\_REGION77.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region77 at reset.

**SAU\_REGION77.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region77 at reset.

**SAU\_REGION77.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region77 at reset.

**SAU\_REGION77.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region77 at reset.

**SAU\_REGION78.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region78 at reset.

**SAU\_REGION78.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region78 at reset.

**SAU\_REGION78.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region78 at reset.

**SAU\_REGION78.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region78 at reset.

**SAU\_REGION79.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region79 at reset.

**SAU\_REGION79.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region79 at reset.

**SAU\_REGION79.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region79 at reset.

**SAU\_REGION79.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region79 at reset.

**SAU\_REGION8.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region8 at reset.

**SAU\_REGION8.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region8 at reset.

**SAU\_REGION8.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region8 at reset.

**SAU\_REGION8.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region8 at reset.

**SAU\_REGION80.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region80 at reset.

**SAU\_REGION80.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region80 at reset.

**SAU\_REGION80.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region80 at reset.

**SAU\_REGION80.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region80 at reset.

**SAU\_REGION81.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region81 at reset.

**SAU\_REGION81.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region81 at reset.

**SAU\_REGION81.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region81 at reset.

**SAU\_REGION81.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region81 at reset.

**SAU\_REGION82.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region82 at reset.

**SAU\_REGION82.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region82 at reset.

**SAU\_REGION82.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region82 at reset.

**SAU\_REGION82.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region82 at reset.

**SAU\_REGION83.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region83 at reset.

**SAU\_REGION83.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region83 at reset.

**SAU\_REGION83.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region83 at reset.

**SAU\_REGION83.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region83 at reset.

**SAU\_REGION84.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region84 at reset.

**SAU\_REGION84.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region84 at reset.

**SAU\_REGION84.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region84 at reset.

**SAU\_REGION84.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region84 at reset.

**SAU\_REGION85.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region85 at reset.

**SAU\_REGION85.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region85 at reset.



**SAU\_REGION85.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region85 at reset.

**SAU\_REGION85.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region85 at reset.

**SAU\_REGION86.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region86 at reset.

**SAU\_REGION86.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region86 at reset.

**SAU\_REGION86.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region86 at reset.

**SAU\_REGION86.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region86 at reset.

**SAU\_REGION87.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region87 at reset.

**SAU\_REGION87.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region87 at reset.

**SAU\_REGION87.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region87 at reset.

**SAU\_REGION87.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region87 at reset.

**SAU\_REGION88.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region88 at reset.

**SAU\_REGION88.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region88 at reset.

**SAU\_REGION88.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region88 at reset.

**SAU\_REGION88.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region88 at reset.

**SAU\_REGION89.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region89 at reset.

**SAU\_REGION89.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region89 at reset.

**SAU\_REGION89.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region89 at reset.

**SAU\_REGION89.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region89 at reset.

**SAU\_REGION9.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region9 at reset.

**SAU\_REGION9.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region9 at reset.

**SAU\_REGION9.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region9 at reset.

**SAU\_REGION9.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region9 at reset.

**SAU\_REGION90.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region90 at reset.

**SAU\_REGION90.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region90 at reset.

**SAU\_REGION90.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region90 at reset.

**SAU\_REGION90.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region90 at reset.

**SAU\_REGION91.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region91 at reset.

**SAU\_REGION91.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region91 at reset.

**SAU\_REGION91.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region91 at reset.

**SAU\_REGION91.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region91 at reset.

**SAU\_REGION92.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region92 at reset.

**SAU\_REGION92.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region92 at reset.

**SAU\_REGION92.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region92 at reset.

**SAU\_REGION92.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region92 at reset.

**SAU\_REGION93.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region93 at reset.

**SAU\_REGION93.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region93 at reset.

**SAU\_REGION93.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region93 at reset.

**SAU\_REGION93.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region93 at reset.

**SAU\_REGION94.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region94 at reset.

**SAU\_REGION94.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region94 at reset.



**SAU\_REGION94.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region94 at reset.

**SAU\_REGION94.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region94 at reset.

**SAU\_REGION95.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region95 at reset.

**SAU\_REGION95.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region95 at reset.

**SAU\_REGION95.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region95 at reset.

**SAU\_REGION95.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region95 at reset.

**SAU\_REGION96.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region96 at reset.

**SAU\_REGION96.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region96 at reset.

**SAU\_REGION96.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region96 at reset.

**SAU\_REGION96.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region96 at reset.

**SAU\_REGION97.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region97 at reset.

**SAU\_REGION97.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region97 at reset.

**SAU\_REGION97.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region97 at reset.

**SAU\_REGION97.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region97 at reset.

**SAU\_REGION98.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region98 at reset.

**SAU\_REGION98.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region98 at reset.

**SAU\_REGION98.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region98 at reset.

**SAU\_REGION98.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region98 at reset.

**SAU\_REGION99.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region99 at reset.

**SAU\_REGION99.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region99 at reset.

**SAU\_REGION99.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region99 at reset.

**SAU\_REGION99.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region99 at reset.

**SAU\_TYPE.SREGION****Type**

int

**Default value**

0x10

**Description**

Number of SAU regions (0 =&gt; no SAU).

**SECEXT****Type**

bool

**Default value**

0x1

**Description**

Whether the ARMv8-M Security Extensions are included.

**SYST****Type**

int

**Default value**

0x2

**Description**

Include SysTick timer functionality (0=Absent, 1=Secure only, 2=Secure and NS).

**SYST\_CALIB\_NS\_reset****Type**

int

**Default value**

0x0

**Description**

SYST\_CALIB\_NS reset value.

**SYST\_CALIB\_reset****Type**

int

**Default value**

0x0

**Description**

SYST\_CALIB reset value.

**VTOR\_NS****Type**

bool

**Default value**

0x1

**Description**

NonSecure Vector Table Offset Register is writeable.

**VTOR\_NS\_MASK****Type**

int

**Default value**

0xffffffff80

**Description**

Non-Secure VTOR write mask.

**VTOR\_S****Type**

bool

**Default value**

0x1

**Description**

Secure Vector Table Offset Register is writeable.

**VTOR\_S\_MASK****Type**

int

**Default value**

0xffffffff80

**Description**

Secure VTOR write mask.

**WIC****Type**

bool

**Default value**

0x1

**Description**

Include support for WIC-mode deep sleep.

**abort\_unaligned\_nonNormal****Type**

bool

**Default value**

0x1

**Description**

If true, UNPREDICTABLE accesses of device and strongly ordered memory abort; if false they are allowed.

**aircr\_iesb\_is\_writable****Type**

bool

**Default value**

0x1

**Description**

IS the AIRCR.IESB bit [5] writable?.

**aircr\_iesb\_reset****Type**

bool

**Default value**

0x0

**Description**

Set the AIRCR.IESB bit [5] after reset.

**`allow_dap_writes_while_core_running`****Type**

bool

**Default value**

0x1

**Description**

Debug writes are respected even while the core is running, i.e. the core does not have to be halted.

**`allow_debug_monitor_with_in_flight_inst`****Type**

bool

**Default value**

0x0

**Description**

Allow handling Debug Monitor exception with in-flight instructions.

**`allow_stack_accesses_to_ppb_space`****Type**

bool

**Default value**

0x0

**Description**

Allow stack accesses to PPB space.

**`always_undefinstr_over_nocp`****Type**

bool

**Default value**

0x0

**Description**

Only v8.0M. Always fault with UNDEFINSTR for undefined instructions that fall in CP space (don't check coprocessor status).

**`apply_prigroup_to_pending_tree`****Type**

bool



**Default value**

0x0

**Description**

DEPRECATED: please use `sep_sec_state_then_apply_prigroup_to_pending`. Original description: Apply AIRCR.PRIGROUP to the pending and active trees (instead of just the active tree) when selecting the highest priority pending exception.

**baseline****Type**

bool

**Default value**

0x1

**Description**

Use the baseline profile (if false, use mainline).

**bp\_on\_2nd\_halfword****Type**

bool

**Default value**

0x1

**Description**

Respect DWT/BPU breakpoint-hit on 2nd halfword of 32-bit instruction.

**callee\_register\_push\_low\_to\_high****Type**

bool

**Default value**

0x1

**Description**

If true, push callee registers in order from R4 to R11. If false, push R11 to R4.

**cde\_fp\_check\_on\_unsupported****Type**

bool

**Default value**

0x0

**Description**

Run FP checks on both supported and unsupported CDE instructions.

**cde\_impl\_name****Type**

string

**Default value**

""

**Description**

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

**clear\_non\_secure\_EXC\_RETURN.ES\_on\_tailchain****Type**

bool

**Default value**

0x1

**Description**

Clear EXC\_RETURN.ES in LR value on entry to a tail-chained exception when returning from Non-secure state.

**condition\_flags\_reset****Type**

int

**Default value**

0x0

**Description**

Reset Value of condition flags in APSR.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**cpu\_can\_access\_debug\_regs****Type**

bool

**Default value**

0x1

**Description**

The DWT, BPU, ROM table, DCB, and the SHCSR and DFSR registers access from the processor.

**dbg\_coproc\_load\_store\_enable****Type**

int

**Default value**

0x0

**Description**

Enable LDCX and STCX instructions.

**dcache-invalidate-ns-cleans-s****Type**

bool

**Default value**

0x0

**Description**

Whether V8M DCI\* in non-secure should clean-and-invalidate secure cache contents.

**dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-cache size in bytes.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**delay\_faultmask\_update****Type**

bool

**Default value**

0x0

**Description**

Delay FAULTMASK update to context sync.

**delay\_sysreg\_update****Type**

bool

**Default value**

0x0

**Description**

Delay some system register updates (e.g. SHCSR) to context sync.

**do\_exclusive\_monitor\_check\_first****Type**

bool

**Default value**

0x0

**Description**

In exclusive stores, check local exclusive monitor before detecting other memory aborts.

**drop\_mem\_fault****Type**

bool

**Default value**

0x0

**Description**

Whether to drop mem\_fault in favour of subsequently generated NOCP/secure fault in PushStack.

**dtcm\_enable****Type**

bool

**Default value**

0x0

**Description**

Enable DTCM at reset.

**dtcm\_size****Type**

int

**Default value**

0x100

**Description**

DTCM size in KB.

**dwt\_unaligned\_word\_access\_as\_half\_word****Type**

bool

**Default value**

0x1

**Description**

DWT Treat unaligned word access as half word or bytes.

**enable\_helium\_extension****Type**

bool

**Default value**

0x0

**Description**

Enable Helium extension.

**exercise\_strex\_fail****Type**

bool

**Default value**

0x0

**Description**

Reject a pseudo-random majority of exclusive store instructions.

**has\_ahbp****Type**

bool

**Default value**

0x1

**Description**

Are Vendor-Sys accesses sent to a separate bus (AHBP on CM7).

**has\_arm\_v8-1m****Type**

bool

**Default value**

0x0

**Description**

Enable v8.1M architecture version and features.

**has\_cde****Type**

bool

**Default value**

0x0

**Description**

Enables Custom Datapath Extensions.

**has\_core\_dside\_bus\_gasket****Type**

bool

**Default value**

0x0

**Description**

STL gasket enabled.

**has\_lob\_cache****Type**

bool

**Default value**

0x1

**Description**

Support for LOB cache (only if support for LO instructions is enabled as well).

**has\_m55\_tcmcr****Type**

bool

**Default value**

0x0

**Description**

If true, enables the CortexM55 TCM Control Registers (ITCMCR and DTCMCR), If false, CortexM55 TCM Control Registers are disabled.

**has\_pmu****Type**

bool

**Default value**

0x0

**Description**

Availability of optional PMU.

**has\_separate\_etm\_reset****Type**

bool

**Default value**

0x0

**Description**

If true, signal 'etmreset' resets the core, else the core power-on-reset does.

**has\_unpriviledged\_debug****Type**

bool

**Default value**

0x1

**Description**

Unprivileged Debug Extension supported for Mainline Extension.

**has\_writebuffer****Type**

bool

**Default value**

0x0

**Description**

Implement write accesses buffering before L1 cache. May affect ext\_abort behaviour.

**icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**ignore\_RNR\_top\_nibble****Type**

bool

**Default value**

0x0

**Description**

If set, only the bottom four bits of MPU\_RNR.REGION are used.

**ignore\_demcr\_sdme\_for\_nonhalting\_bkpt****Type**

bool

**Default value**

0x0

**Description**

Ignore the SDME bit of the DEMCR register when escalating a Debug Monitor exception to a HardFault.

**ignore\_out\_of\_range\_RNR\_write****Type**

bool



**Default value**

0x0

**Description**

If an MPU\_RNR.REGION write is out of range, ignore it ; if false, MPU\_RNR values wrap.

**ignore\_unpred\_SBZSBO****Type**

bool

**Default value**

0x0

**Description**

Use smaller decoder does not UNDEF some unpredictable SBZ/SBO fields.

**ignore\_unpred\_ZeroRegistersInList****Type**

bool

**Default value**

0x0

**Description**

VLDM,VSTM,STM,LDM with no registers NOP instead of UNDEF.

**itcm\_enable****Type**

bool

**Default value**

0x0

**Description**

Enable ITCM at reset.

**itcm\_size****Type**

int

**Default value**

0x100

**Description**

ITCM size in KB.

**late\_arrival****Type**

bool

**Default value**

0x1

**Description**

Enable late arrival support.

**master\_id****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**mve\_has\_atomic\_ticks****Type**

bool

**Default value**

0x0

**Description**

Enable atomic ticks behaviour for vector instructions flagged as such (e.g. VLDR).

**num\_pmu\_counters****Type**

int

**Default value**

0x1f

**Description**

Number of available PMU counters.

**number\_of\_itm\_stimulus\_ports****Type**

int

**Default value**

0x20

**Description**

The number of ITM stimulus ports.

**pend\_overriden\_exception\_on\_stack\_push****Type**

bool

**Default value**

0x0

**Description**

Mark any overridden exceptions on stack push as pending (instead of dropping them).

**preserve\_unknown\_caller\_save\_regs\_at\_S\_to\_S****Type**

bool

**Default value**

0x1

**Description**

preserve unknown caller registers when they become UNKNOWN at secure to secure.

**ras\_ERRFR0****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR1****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR10****Type**

string

**Default value**

"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR11****Type**

string

**Default value**

"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR12****Type**

string

**Default value**

"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR13****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR14****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR15****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR16****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR17****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR18****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR19****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR2****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR20****Type**

string

**Default value**

""{"ED\\":0x2, \\\"UI\\\":0x2, \\\"FI\\\":0x2, \\\"UE\\\":0x2, \\\"CFI\\\":0x2, \\\"CEC\\\":0x2, \\\"RP\\\":0x1, \\\"DUI\\\":0x0, \\\"CEO\\\":0x1, \\\"CI\\\":0x2}""

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR21****Type**

string

**Default value**

""{"ED\\":0x2, \\\"UI\\\":0x2, \\\"FI\\\":0x2, \\\"UE\\\":0x2, \\\"CFI\\\":0x2, \\\"CEC\\\":0x2, \\\"RP\\\":0x1, \\\"DUI\\\":0x0, \\\"CEO\\\":0x1, \\\"CI\\\":0x2}""

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR22****Type**

string

**Default value**

""{"ED\\":0x2, \\\"UI\\\":0x2, \\\"FI\\\":0x2, \\\"UE\\\":0x2, \\\"CFI\\\":0x2, \\\"CEC\\\":0x2, \\\"RP\\\":0x1, \\\"DUI\\\":0x0, \\\"CEO\\\":0x1, \\\"CI\\\":0x2}""

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR23****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR24****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR25****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR26****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.



**ras\_ERRFR27****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR28****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR29****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR3****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR30****Type**

string

**Default value**

"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR31****Type**

string

**Default value**

"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR32****Type**

string

**Default value**

"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR33****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR34****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR35****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR36****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR37****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR38****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR39****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR4****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR40****Type**

string

**Default value**

"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR41****Type**

string

**Default value**

"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR42****Type**

string

**Default value**

"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR43****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR44****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR45****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR46****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR47****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR48****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR49****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR5****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR50****Type**

string

**Default value**

""{"ED\\":0x2, \\\"UI\\\":0x2, \\\"FI\\\":0x2, \\\"UE\\\":0x2, \\\"CFI\\\":0x2, \\\"CEC\\\":0x2, \\\"RP\\\":0x1, \\\"DUI\\\":0x0, \\\"CEO\\\":0x1, \\\"CI\\\":0x2}""

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR51****Type**

string

**Default value**

""{"ED\\":0x2, \\\"UI\\\":0x2, \\\"FI\\\":0x2, \\\"UE\\\":0x2, \\\"CFI\\\":0x2, \\\"CEC\\\":0x2, \\\"RP\\\":0x1, \\\"DUI\\\":0x0, \\\"CEO\\\":0x1, \\\"CI\\\":0x2}""

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR52****Type**

string

**Default value**

""{"ED\\":0x2, \\\"UI\\\":0x2, \\\"FI\\\":0x2, \\\"UE\\\":0x2, \\\"CFI\\\":0x2, \\\"CEC\\\":0x2, \\\"RP\\\":0x1, \\\"DUI\\\":0x0, \\\"CEO\\\":0x1, \\\"CI\\\":0x2}""

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR53****Type**

string



**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR54****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR55****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR6****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR7****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR8****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_ERRFR9****Type**

string

**Default value**

```
"{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}"
```

**Description**

A JSON object or array of objects for each field of ERRFR. Records not described default to RAZ e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

**ras\_cei\_pin****Type**

int

**Default value**

0x2

**Description**

RAS: Critical error interrupt pin.

**ras\_cei\_support****Type**

bool

**Default value**

0x1

**Description**

RAS: Whether Critical Error Interrupt is supported.

**ras\_eri\_pin****Type**

int

**Default value**

0x1

**Description**

RAS: Error recovery interrupt pin.

**ras\_eri\_support****Type**

bool

**Default value**

0x1

**Description**

RAS: Whether Error Recovery Interrupt is supported.

**ras\_error\_record****Type**

int

**Default value**

0xffffffffffffffff

**Description**

56 bit value that specifies which nodes out of 0-55 are implemented (ERRDEVID is derived from this parameter).

**ras\_fhi\_pin****Type**

int

**Default value**

0x0

**Description**

RAS: Fault handling interrupt pin.

**ras\_fhi\_support****Type**

bool

**Default value**

0x1

**Description**

RAS: Whether Fault Handling Interrupt is supported.

**rd\_ns\_bus\_err\_behave****Type**

int

**Default value**

0x1

**Description**

External read aborts in nonsecure domain 0:ignored, 1:precise, 2:imprecise, 3=imprecise except SO.

**rd\_s\_bus\_err\_behave****Type**

int

**Default value**

0x1

**Description**

External read aborts in secure domain 0:ignored, 1:precise, 2:imprecise, 3=imprecise except SO.

**register\_reset\_data****Type**

int

**Default value**

0x0

**Description**

Data used to fill register bits when they become UNKNOWN at reset.

**reported\_patch\_level****Type**

int

**Default value**`0xffffffffffffffff`**Description**

Purely cosmetic patch level value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**`int`**Default value**`0xffffffffffffffff`**Description**

Purely cosmetic revision number value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**semihosting-Thumb\_SVC****Type**`int`**Default value**`0xab`**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**`string`**Default value**`""`**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**`string`**Default value**`""`**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**semihosting-stack\_base****Type**

int

**Default value**

0x20800000

**Description**

Virtual address of base of descending stack.

**`semihosting-stack_limit`****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of stack limit.

**`sep_sec_state_then_apply_prigroup_to_pending`****Type**

bool

**Default value**

0x0

**Description**

Use separate comparison trees for Secure and Non-Secure pending exceptions and apply AIRCR.PRIGROUP to the output of each before they are compared to determine the overall highest priority.

**`sequential_security_transitions`****Type**

int

**Default value**

0x1

**Description**

Allow transition of security state in sequential instruction fetches that cross from non-secure to secure memory with SG instruction 0: never, 1: always, 2: 32-bit instrs, 3: ISB.

**`share_fault_address_reg`****Type**

bool

**Default value**

0x0

**Description**

If true, Fault Address Register is shared.

**stack\_limit\_check****Type**

int

**Default value**

0x7

**Description**

Implementation defined stack limit checks for instructions. Bit 0: Load-exclusive, Bit 1: Load-acquire, Bit 2: VLDM. Any instruction that can't be configured does stack limit check by default.

**stack\_limit\_check\_optimization****Type**

bool

**Default value**

0x1

**Description**

Stack limit check optimization (0: limit check done for each word on the stack, 1: limit check done only on stack pointer).

**stacking\_writes\_are\_precise****Type**

bool

**Default value**

0x1

**Description**

Faults on stack writes are precise.

**supports\_unprivileged****Type**

bool

**Default value**

0x1

**Description**

Enable support for Unprivileged/Privileged Extension.

**tail\_chain****Type**

bool

**Default value**

0x1



**Description**

Enable tail-chaining optimisation.

**trace\_style****Type**

int

**Default value**

0x2

**Description**

MVE instruction trace style: Add 16 for [\*\*-] beat trace. Add 32 for tracing IMPLIED LOB instructions. Add 64 to change opcode of implied BF to 0xBF00.

**unknown\_regs\_at\_exception\_value****Type**

int

**Default value**

0x0

**Description**

Data used to fill registers when they become UNKNOWN at exception and exception-return.

**unpred\_WriteBackandBaseInList\_stores\_old\_base\_value****Type**

bool

**Default value**

0x0

**Description**

allow STM with write back to base register in register list.

**unpred\_mon\_step\_write****Type**

int

**Default value**

0x0

**Description**

Behavior on unpredictable updates to MON\_STEP bit of DEMCR. 0: ignore write, 1: set one, 2: set zero.

**unpred\_msr\_psr\_with\_one\_mask\_and\_nodsp\_is\_nop****Type**

bool

**Default value**

0x1

**Description**

If true, MSR to \*PSR with a one mask and no DSP does nothing.

**unpred\_msr\_psr\_with\_zero\_mask\_is\_nop****Type**

bool

**Default value**

0x0

**Description**

If true, MSR to \*PSR with a zero mask does nothing.

**unstack\_R\_regs\_before\_fp\_cp\_check****Type**

bool

**Default value**

0x0

**Description**

In exception return unstack normal register before checking fp coprocessor is enable to unstack FP register.

**vector\_fetch\_as\_wpt\_event****Type**

bool

**Default value**

0x0

**Description**

Watchpoint on exception vector fetch.

**vector\_fetch\_busfault\_sets\_HFSR\_FORCED****Type**

bool

**Default value**

0x0

**Description**

Only v8.0M. Set HFSR.FORCED when a vector table read generates a HardFault.

**vector\_fetch\_on\_iside****Type**

bool

**Default value**

0x1

**Description**

Perform vector fetch on I-side.

**vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**wr\_ns\_bus\_err\_behave****Type**

int

**Default value**

0x3

**Description**

External write aborts in nonsecure domain 0:ignored, 1:precise, 2:imprecise, 3=imprecise except SO.

**wr\_s\_bus\_err\_behave****Type**

int

**Default value**

0x3

**Description**

External write aborts in secure domain 0:ignored, 1:precise, 2:imprecise, 3=imprecise except SO.

**write\_unknown\_regs\_at\_exception**

**Type**

bool

**Default value**

0x0

**Description**

Do we write registers when they become UNKNOWN at exception or exception-return.

3.5.4 ARMCortexA5CT

ARMCortexA5CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-195: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexA5CT

- The ase-present and vfp-present parameters configure the synthesis options:
  - vfp present and ase present**  
Neon™ and VFPv3-D32 are supported.
  - vfp present and ase not present**  
VFPv3-D16 is supported.
  - vfp not present and ase present**  
Illegal. Forces vfp-present to true so model has Neon and VFPv3-D32 support.
  - vfp not present and ase not present**  
Model has neither Neon nor VFPv3-D32 support.
- Specify false for the device-accurate-tlb parameter to enable modeling a different number of TLBs if this improves simulation performance. In this case, the simulation is architecturally-accurate, but not device-accurate. Architectural accuracy is almost always sufficient. Specify true if device accuracy is required.

Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- The RR bit in the SCTLR is ignored.
- The Power Control Register in the system control coprocessor is implemented but writing to it does not change the behavior of the model.
- This component implements L1 cache as architecturally defined, but does not implement L2 cache. If you require an L2 cache you can add a PL310 Level 2 Cache Controller component.

### Iris and MTI instances for ARMCortexA5CT

This model has the following Iris instances:

**Table 3-196: ARMCortexA5CT Iris instances**

InstanceName	ComponentName
ARMCortexA5CT	Cluster_ARM_Cortex-A5UP
ARMCortexA5CT.acp_mapper	PVBusMapper
ARMCortexA5CT.cpu0	ARM_Cortex-A5UP
ARMCortexA5CT.cpu0.UTLB	TLB
ARMCortexA5CT.cpu0.l1dcache	PVCache
ARMCortexA5CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA5CT.cpu0.l1icache	PVCache
ARMCortexA5CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA5CT.cpu0.utlb	TLB
ARMCortexA5CT.ext_bus	PVBusLogger
ARMCortexA5CT.ext_bus.mapper	PVBusMapper
ARMCortexA5CT.l1_incoherent_interconnect	PVCache
ARMCortexA5CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave

InstanceName	ComponentName
ARMCortexA5CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-197: ARMCortexA5CT MTI instances**

InstanceName	ComponentName
ARMCortexA5CT.acp_mapper	PVBusMapper
ARMCortexA5CT.cpu0	ARM_Cortex-A5
ARMCortexA5CT.cpu0.UTLB	TLB
ARMCortexA5CT.cpu0.l1dcache	PVCache
ARMCortexA5CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA5CT.cpu0.l1icache	PVCache
ARMCortexA5CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA5CT.ext_bus	PVBusLogger
ARMCortexA5CT.ext_bus.mapper	PVBusMapper
ARMCortexA5CT.l1_incoherent_interconnect	PVCache
ARMCortexA5CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexA5CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexA5CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA5CT

**Table 3-198: Ports**

Name	Protocol	Type	Description
cfgend[1]	Signal	Slave	This signal is for EE bit initialisation.

Name	Protocol	Type	Description
cfgnmfi[1]	Signal	Slave	This signal disables FIQ mask in CPSR.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
cp15sdisable[1]	Signal	Slave	This signal disables write access to some system control processor registers.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[1]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
irq[1]	Signal	Slave	This signal drives the CPU's interrupt handling.
pmuirq[1]	Signal	Master	Interrupt signal from performance monitoring unit.
pvbuse_m0	PVBus	Master	The core will generate bus requests on this port.
reset[1]	Signal	Slave	Raising this signal will put the core into reset mode.
standbywfe[1]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[1]	Signal	Master	This signal indicates if a core is in WFI state.
teinit[1]	Signal	Slave	This signal provides default exception handling state.
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vinithi[1]	Signal	Slave	This signal controls of the location of the exception vectors at reset.

## Parameters for ARM Cortex A5CT

### CLUSTER\_ID

#### Type

int

#### Default value

0x0

#### Description

Processor cluster ID value.

### cpi\_div

#### Type

int

#### Default value

0x1

#### Description

Divider for calculating CPI (Cycles Per Instruction).

### cpi\_mul

#### Type

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Initialize to BE8 endianness.

**cpuX.CFGNMFI****Type**

bool

**Default value**

0x0

**Description**

Enable nonmaskable FIQ interrupts on startup.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.POWERCTLI****Type**

int

**Default value**

0x0

**Description**

Default power control state for processor.

**cpuX.TEINIT****Type**

bool



**Default value**

0x0

**Description**

T32 exception enable. The default has exceptions including reset handled in A32 state.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether model has NEON support.

**cpuX.dcache-size****Type**

int

**Default value**

0x8000

**Description**

Set D-cache size in bytes.

**cpuX.icache-size****Type**

int

**Default value**

0x8000

**Description**

Set I-cache size in bytes.

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-ARM\_HLT****Type**

int

**Default value**

0xf000

**Description**

ARM HLT number for semihosting.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

ARM SVC number for semihosting.

**cpuX.semihosting-Thumb\_HLT****Type**

int

**Default value**

0x3c

**Description**

Thumb HLT number for semihosting.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

Thumb SVC number for semihosting.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-hlt-enable****Type**

bool

**Default value**

0x0

**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable coprocessor access and VFP at reset.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**device-accurate-tlb****Type**

bool

**Default value**

0x0

**Description**

Specify whether all TLBs are modeled.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

I-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.



### 3.5.5 ARMCortexA5MPx1CT

ARMCortexA5MPx1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-199: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About ARMCortexA5MPx1CT

- The following components also exist:
  - ARMCortexA5MPx2CT.
  - ARMCortexA5MPx4CT.

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-3).

- The `ase-present` and `vfp-present` parameters configure the synthesis options:

##### **vfp present and ase present**

Neon™ and VFPv3-D32 are supported.

##### **vfp present and ase not present**

VFPv3-D16 is supported.

##### **vfp not present and ase present**

Illegal. Forces `vfp-present` to true so model has Neon and VFPv3-D32 support.

##### **vfp not present and ase not present**

Model has neither Neon nor VFPv3-D32 support.

- Specify `false` for the `device-accurate-tlb` parameter to enable modeling a different number of TLBs if this improves simulation performance. In this case, the simulation is architecturally-accurate, but not device-accurate. Architectural accuracy is almost always sufficient. Specify `true` if device accuracy is required.
- If you are using the ARMCortexA5MPx<sub>n</sub>CT component in a VE platform model, the `PERIPHBASE` parameter is set automatically to `0x2C000000` and is not visible in the parameter list.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to `false` in production systems.

#### Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- There is a single memory port combining instruction, data, DMA and peripheral access.
- The GIC does not respect the `CFGSDISABLE` signal. This leads to some registers being accessible when they must not be.

- The SCU enable bit is ignored. The SCU is always enabled.
- The SCU ignores the invalidate all register.
- The Broadcast TLB or cache operations in this model do not cause other cores in the cluster that are asleep because of *Wait For Interrupt* (WFI) to wake up.
- The RR bit in the SCTLR is ignored.
- The Power Control Register in the system control coprocessor is implemented but writing to it does not change the behavior of the model.
- When modeling the SCU, coherency operations are represented by a memory write followed by a read to refill from memory, rather than using cache-to-cache transfers.
- This component implements L1 cache as architecturally defined, but does not implement L2 cache. If you require an L2 cache you can add a PL310 Level 2 Cache Controller component.

### Iris and MTI instances for ARMCortexA5MPx1CT

This model has the following Iris instances:

**Table 3-200: ARMCortexA5MPx1CT Iris instances**

InstanceName	ComponentName
ARMCortexA5MPx1CT	Cluster_ARM_Cortex-A5MP
ARMCortexA5MPx1CT.acp_mapper	PVBusMapper
ARMCortexA5MPx1CT.cpu0	ARM_Cortex-A5MP
ARMCortexA5MPx1CT.cpu0.UTLB	TLB
ARMCortexA5MPx1CT.cpu0.l1dcache	PVCache
ARMCortexA5MPx1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA5MPx1CT.cpu0.l1licache	PVCache
ARMCortexA5MPx1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA5MPx1CT.cpu0.utlb	TLB
ARMCortexA5MPx1CT.ext_bus	PVBusLogger
ARMCortexA5MPx1CT.ext_bus.mapper	PVBusMapper
ARMCortexA5MPx1CT.internal_shareability_remapper	PVBusMapper
ARMCortexA5MPx1CT.l1_incoherent_interconnect	PVCache
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave

InstanceName	ComponentName
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexA5MPx1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-201: ARMCortexA5MPx1CT MTI instances**

InstanceName	ComponentName
ARMCortexA5MPx1CT.acp_mapper	PVBusMapper
ARMCortexA5MPx1CT.cpu0	ARM_Cortex-A5MP
ARMCortexA5MPx1CT.cpu0.UTLB	TLB
ARMCortexA5MPx1CT.cpu0.l1dcache	PVCache
ARMCortexA5MPx1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA5MPx1CT.cpu0.l1licache	PVCache
ARMCortexA5MPx1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA5MPx1CT.ext_bus	PVBusLogger
ARMCortexA5MPx1CT.ext_bus.mapper	PVBusMapper
ARMCortexA5MPx1CT.internal_shareability_remapper	PVBusMapper
ARMCortexA5MPx1CT.l1_incoherent_interconnect	PVCache
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave

InstanceName	ComponentName
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexA5MPx1CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexA5MPx1CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA5MPx1CT

**Table 3-202: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
cfgend[1]	Signal	Slave	This signal is for EE bit initialisation.
cfgnmfi[1]	Signal	Slave	This signal disables FIQ mask in CPSR.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
cp15sdisable[1]	Signal	Slave	This signal disables write access to some system control processor registers.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
filteren	Signal	Slave	This signal enables filtering of address ranges between master bus ports.
filterend	Value	Slave	This port sets end of region mapped to pvbus_m1.
filterstart	Value	Slave	This port sets start of region mapped to pvbus_m1.
fiq[1]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
fiqout[1]	Signal	Master	This signal exports internal VGIC FIQ signal to the CPU.
ints[224]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
irq[1]	Signal	Slave	This signal drives the CPU's interrupt handling.
irqout[1]	Signal	Master	This signal exports internal VGIC IRQ signal to the CPU.
periphbase	Value	Slave	This port sets the base address of private peripheral region.
periphclk_in	ClockSignal	Slave	The timer and the watchdog take need a clk that is scaled down at least by factor of two.
periphreset	Signal	Slave	This signal resets timer and interrupt controller.
pmuirq[1]	Signal	Master	Interrupt signal from performance monitoring unit.
pvbus_m0	PVBus	Master	AXI master 0 bus master channel.
pvbus_m1	PVBus	Master	AXI master 1 bus master channel.
pwrctl1[1]	Value	Slave	This port sets reset value for scu CPU status register.
pwrctl0[1]	Value	Master	This port sends scu CPU status register bits.
reset[1]	Signal	Slave	Raising this signal will put the core into reset mode.
scureset	Signal	Slave	This signal resets SCU.
smpnamp[1]	Signal	Master	This signals AMP or SMP mode for each Cortex-A5 processor.
standbywfe[1]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[1]	Signal	Master	This signal indicates if a core is in WFI state.
teinit[1]	Signal	Slave	This signal provides default exception handling state.

Name	Protocol	Type	Description
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vinithi[1]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
wdreset[1]	Signal	Slave	This signal resets individual watchdog.
wdresetreq[1]	Signal	Master	This signal resets rest of the CA5MP system.

## Parameters for ARM Cortex A5MPx1CT

### CFGSDISABLE

#### Type

bool

#### Default value

0x0

#### Description

Disable some accesses to GIC registers.

### CLUSTER\_ID

#### Type

int

#### Default value

0x0

#### Description

Processor cluster ID value.

### FILTEREN

#### Type

bool

#### Default value

0x0

#### Description

Enable filtering of accesses through pvbus\_m0.

### FILTEREND

#### Type

int

#### Default value

0x0

**Description**

End of region filtered to pvbus\_m0.

**FILTERSTART****Type**

int

**Default value**

0x0

**Description**

Base of region filtered to pvbus\_m0.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Initialize to BE8 endianness.

**cpuX.CFGNMFI****Type**

bool

**Default value**

0x0

**Description**

Enable nonmaskable FIQ interrupts on startup.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.POWERCTLI****Type**

int

**Default value**

0x0

**Description**

Default power control state for processor.

**cpuX.SMPnAMP****Type**

bool

**Default value**

0x0

**Description**

Set whether the processor is part of a coherent domain.

**cpuX.TEINIT****Type**

bool

**Default value**

0x0

**Description**

T32 exception enable. The default has exceptions including reset handled in A32 state.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether model has NEON support.

**cpuX.dcache-size****Type**

int

**Default value**

0x8000

**Description**

Set D-cache size in bytes.

**cpuX.icache-size****Type**

int

**Default value**

0x8000

**Description**

Set I-cache size in bytes.

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0



**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-ARM\_HLT****Type**

int

**Default value**

0xf000

**Description**

ARM HLT number for semihosting.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

ARM SVC number for semihosting.

**cpuX.semihosting-Thumb\_HLT****Type**

int

**Default value**

0x3c

**Description**

Thumb HLT number for semihosting.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

Thumb SVC number for semihosting.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-hlt-enable****Type**

bool

**Default value**

0x0

**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable coprocessor access and VFP at reset.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**device-accurate-tlb****Type**

bool

**Default value**

0x0

**Description**

Specify whether all TLBs are modeled.

**dic-spi\_count****Type**

int

**Default value**

0x40

**Description**

Number of shared peripheral interrupts implemented.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

I-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-state\_modelled**

**Type**  
bool

**Default value**  
0x0

**Description**  
Set whether I-cache has stateful implementation.

3.5.6 **ARMCortexA7x1CT**

ARMCortexA7x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-203: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About **ARMCortexA7x1CT**

- The following components also exist:
    - ARMCortexA7x2CT.
    - ARMCortexA7x3CT.
    - ARMCortexA7x4CT.
- The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-3).
- The `ase-present` and `vfp-present` parameters configure the synthesis options:
    - vfp present and ase present**  
Neon™ and VFPv4-D32 are supported.
    - vfp present and ase not present**  
VFPv4-D16 is supported.
    - vfp not present and ase present**  
Illegal. Forces `vfp-present` to true so model has Neon and VFPv4-D32 support.
    - vfp not present and ase not present**  
Model has neither Neon nor VFPv4-D32 support.
  - If you are using the `ARMCortexA7xnCT` component in a VE platform model, the `PERIPHBASE` parameter is set automatically to `0x2C000000` and is not visible in the parameter list.
  - The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to `false` in production systems.



## ACE limitation

AXI Coherency Extensions (ACE) are extensions to AXI4 that support system-level cache coherency between multiple clusters. The ACE cache models in the Arm® Cortex®-A15 and Cortex-A7, and the ACE support in the CCI-400 have the limitation that they only process one transaction at a time. Normally, the simulation processes each transaction to completion before allowing any master to generate another transaction. However, in the following situation the simulation might fail. If a SystemC bus slave calls `wait()` while it is processing a transaction, this call might allow another master to issue another transaction that passes through the CCI-400 or the Cortex-A15 or Cortex-A7 caches. This situation could happen if a SystemC bus master running in another thread is connected to one of the ACE-lite ports on the CCI-400.

## Differences between the CT model and RTL implementations

This model has the following differences from the corresponding revision of the RTL implementation:

- The GIC does not respect the CFGSDISABLE signal. This leads to some registers wrongly being accessible.
- The Broadcast Translation Lookaside Buffer (TLB) or cache operations in this model do not cause other cores in the cluster that are asleep because of Wait For Interrupt (WFI) to wake up.
- The model ignores the RR bit in the SCTLR.
- The model implements the Power Control Register in the system control coprocessor but writing to it does not change the behavior of the model.
- The model does not implement ETM registers.
- The model does not support the Cortex®-A7 mechanism to read the internal memory that the Cache and TLB structures use through the implementation defined region of the system coprocessor interface.
- The model does not upgrade DCIMVAC operations to DCCIMVAC.

## Iris and MTI instances for ARM Cortex A7x1CT

This model has the following Iris instances:

**Table 3-204: ARM Cortex A7x1CT Iris instances**

InstanceName	ComponentName
ARM Cortex A7x1CT	Cluster_ARM_Cortex-A7
ARM Cortex A7x1CT.ARM Cortex A7x1CT.debug_rom	debug_rom
ARM Cortex A7x1CT.Cortex-A7_GIC	GICv2
ARM Cortex A7x1CT.acp_mapper	PVBusMapper
ARM Cortex A7x1CT.cpu0	ARM_Cortex-A7
ARM Cortex A7x1CT.cpu0.DTLB	TLB
ARM Cortex A7x1CT.cpu0.ITLB	TLB
ARM Cortex A7x1CT.cpu0.dtlb	TLB
ARM Cortex A7x1CT.cpu0.itlb	TLB
ARM Cortex A7x1CT.cpu0.l1dcache	PVCache

InstanceName	ComponentName
ARMCortexA7x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA7x1CT.cpu0.l1licache	PVCache
ARMCortexA7x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA7x1CT.ext_bus	PVBusLogger
ARMCortexA7x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA7x1CT.l2_cache	PVCache
ARMCortexA7x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA7x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-205: ARMCortexA7x1CT MTI instances**

InstanceName	ComponentName
ARMCortexA7x1CT.Cortex-A7_GIC	GICv2
ARMCortexA7x1CT.acp_mapper	PVBusMapper
ARMCortexA7x1CT.cpu0	ARM_Cortex-A7
ARMCortexA7x1CT.cpu0.DTLB	TLB
ARMCortexA7x1CT.cpu0.ITLB	TLB
ARMCortexA7x1CT.cpu0.l1dcache	PVCache
ARMCortexA7x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA7x1CT.cpu0.l1licache	PVCache
ARMCortexA7x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA7x1CT.ext_bus	PVBusLogger
ARMCortexA7x1CT.ext_bus.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA7x1CT.l2_cache	PVCache
ARMCortexA7x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA7x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA7x1CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA7x1CT

**Table 3-206: Ports**

Name	Protocol	Type	Description
axierrirq	Signal	Master	Imprecise aborts from the L2 are signaled by pulsing this pin, typically they are connect to an interrupt controller.
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	Signal	Slave	Enable broadcasting of Inner Shareable transactions.
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
cfgend[4]	Signal	Slave	This signal controls the SCTL.R.EE bit.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	This port sets the value in the CLUSTERID field (bits[11:8]) of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Outputs of the generic timers.
CNTPNSIRQ[4]	Signal	Master	Outputs of the generic timers.
CNTPSIRQ[4]	Signal	Master	Outputs of the generic timers.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Outputs of the generic timers.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some secure system control processor registers.

Name	Protocol	Type	Description
cpuporeset[4]	Signal	Slave	Signal initializes all processor logic including NEON, VFP, Debug, PTM, breakpoint and watchpoint.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
fiqout[4]	Signal	Master	This signal exports internal VGIC FIQ signal to the CPU..
irq[4]	Signal	Slave	This signal drives the CPU's interrupt handling.
irqout[4]	Signal	Master	This signal exports internal VGIC IRQ signal to the CPU.
irqs[480]	Signal	Slave	These signals drive the CPU's interrupt controller interrupt lines.
l2reset	Signal	Slave	This signal resets the shared L2 memory system, interrupt controller and timer logic.
periphbase	Value_64	Slave	This port sets the base address of the private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Signal initializes the shared Debug APB, CTI and CTM logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
standbywfi12	Signal	Master	Indicate that all the individual processors and the L2 memory system are in a WFI state.
teinit[4]	Signal	Slave	This signal enables Thumb exceptions (controls the SCTLR.TE bit).
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vfiq[4]	Signal	Slave	Virtual FIQ inputs. Note that the fiq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
vinithi[4]	Signal	Slave	This signal controls the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ inputs. Note that the irq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).

## Parameters for ARM Cortex A7x1CT

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTINNER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**CFGSDISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable some accesses to GIC registers.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

Processor cluster ID value.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Initialize to BE8 endianness.

**`cpuX.CP15SDISABLE`****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**`cpuX.DBGROMADDR`****Type**

int

**Default value**

0x12000003

**Description**

This value is used to initialize the CP15 DBGDRAR register. Bits[39:12] of this register specify the ROM table physical address.

**cpuX.DBGROMADDRV****Type**

bool

**Default value**

0x1

**Description**

If true this sets bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**cpuX.DBGSELFADDR****Type**

int

**Default value**

0x10003

**Description**

This value is used to initialize the CP15 DBGDSAR register. Bits[39:17] of this register specify the ROM table physical address.

**cpuX.DBGSELFADDRV****Type**

bool

**Default value**

0x1

**Description**

If true this sets bits[1:0] of the CP15 DBGDSAR to indicate that the address is valid.

**cpuX.TEINIT****Type**

bool

**Default value**

0x0

**Description**

T32 exception enable. The default has exceptions including reset handled in A32 state.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether CT model has been built with NEON support.

**cpuX.l1\_dcache-size****Type**

int

**Default value**

0x8000

**Description**

Size of L1 D-cache.

**cpuX.l1\_icache-size****Type**

int

**Default value**

0x8000

**Description**

Size of L1 I-cache.

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-ARM\_HLT****Type**

int



**Default value**`0xf000`**Description**

ARM HLT number for semihosting.

**`cpuX.semihosting-ARM_SVC`****Type**`int`**Default value**`0x123456`**Description**

ARM SVC number for semihosting.

**`cpuX.semihosting-Thumb_HLT`****Type**`int`**Default value**`0x3c`**Description**

Thumb HLT number for semihosting.

**`cpuX.semihosting-Thumb_SVC`****Type**`int`**Default value**`0xab`**Description**

Thumb SVC number for semihosting.

**`cpuX.semihosting-cmd_line`****Type**`string`**Default value**`""`**Description**

Command line available to semihosting SVC calls.

**`cpuX.semihosting-cwd`****Type**`string`

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-hlt-enable****Type**

bool

**Default value**

0x0

**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable coprocessor access and VFP at reset.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether CT model has been built with VFP support.

**dic-spi\_count****Type**

int

**Default value**

0x40

**Description**

Number of shared peripheral interrupts implemented.

**disable\_periph\_decoder****Type**

bool

**Default value**

0x0

**Description**

Disable memory mapped access to gic system registers.

**internal\_vgic****Type**

bool

**Default value**

0x1

**Description**

Configures whether the model of the processor contains a VGIC.

**l1\_dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l1\_dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l1\_dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l1\_dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l1_dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l1\_dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `l1_dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**l1\_dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**l1\_dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether L1 D-cache has stateful implementation.

**l1\_dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l1_dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**l1\_dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `l1_dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**l1\_icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`l1_icache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`l1_icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`l1_icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l1_icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`l1_icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `l1_ichache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l1_ichache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether L1 I-cache has stateful implementation.

**`l2_cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`l2_cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`l2_cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.



**l2\_cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2\_cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l2\_cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2\_cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l2\_cache-size****Type**

int

**Default value**

0x80000

**Description**

Set L2 cache size in bytes.

**l2\_cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l2\_cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l2\_cache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether L2 cache has stateful implementation.

**l2\_cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2\_cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l2\_cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2\_cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

### 3.5.7 ARMCortexA8CT

ARMCortexA8CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-207: IP revisions support**

Revision	Quality level
r2p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About ARMCortexA8CT

- Specify `false` for the `device-accurate-tlb` parameter to enable modeling a different number of TLBs if this improves simulation performance. In this case, the simulation is architecturally-accurate, but not device-accurate. Architectural accuracy is almost always sufficient. Specify `true` if device accuracy is required.
- If the `l1_dcache-state_modelled` parameter is `true`, then `l2_cache-state_modelled` must also be `true`.
- When you use the `semihosting-cmd_line` parameter to set the command line that is available to semihosting SVC calls, the value of `argv[0]` is set to the first command-line argument, not to the name of an image.
- This component provides the registers that the Technical Reference Manual (TRM) specifies, except for the coprocessor 14 registers, the integration and test registers, and the PLE model, which is register-based and has no implemented behavior.

These TLB registers do not have working implementations:

- D-TLB ATTR read/write.
- D-TLB CAM read/write.
- D-TLB PA read/write.
- Normal memory remap register.
- Primary memory remap register.

#### Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- There is a single memory port combining instruction, data, DMA and peripheral access.
- The L2 cache write allocate policy is not configurable. It defaults to write-allocate. Writes to the configuration register succeed but are ignored, meaning that data can be unexpectedly stored in the L2 cache.
- Unaligned accesses with the MMU disabled on the processor do not cause data aborts.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to `false` in production systems.

- ECC and parity schemes are hardware-specific so are not supported.

### Iris and MTI instances for ARM Cortex-A8CT

This model has the following Iris instances:

**Table 3-208: ARM Cortex-A8CT Iris instances**

InstanceName	ComponentName
ARM Cortex-A8CT	ARM_Cortex-A8
ARM Cortex-A8CT.DTLB	TLB
ARM Cortex-A8CT.ITLB	TLB
ARM Cortex-A8CT.acp_mapper	PVBusMapper
ARM Cortex-A8CT.cpu0.dtlb	TLB
ARM Cortex-A8CT.cpu0.itlb	TLB
ARM Cortex-A8CT.cpu0.l1dcache	PVCache
ARM Cortex-A8CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM Cortex-A8CT.cpu0.l1licache	PVCache
ARM Cortex-A8CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARM Cortex-A8CT.ext_bus	PVBusLogger
ARM Cortex-A8CT.ext_bus.mapper	PVBusMapper
ARM Cortex-A8CT.l2_cache	PVCache
ARM Cortex-A8CT.l2_cache.upstream[0]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[10]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[11]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[12]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[13]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[14]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[15]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[16]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[1]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[2]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[3]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[4]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[5]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[6]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[7]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[8]	PVBusSlave
ARM Cortex-A8CT.l2_cache.upstream[9]	PVBusSlave
ARM Cortex-A8CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-209: ARMCortexA8CT MTI instances**

InstanceName	ComponentName
ARMCortexA8CT	ARM_Cortex-A8
ARMCortexA8CT.DTLB	TLB
ARMCortexA8CT.ITLB	TLB
ARMCortexA8CT.acp_mapper	PVBusMapper
ARMCortexA8CT.cpu0.l1dcache	PVCache
ARMCortexA8CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA8CT.cpu0.l1icache	PVCache
ARMCortexA8CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA8CT.ext_bus	PVBusLogger
ARMCortexA8CT.ext_bus.mapper	PVBusMapper
ARMCortexA8CT.l2_cache	PVCache
ARMCortexA8CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA8CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA8CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA8CT

**Table 3-210: Ports**

Name	Protocol	Type	Description
cfgend0	Signal	Slave	Configure BE8 mode after a reset.
cfgnmfi	Signal	Slave	Configure FIQs as non-maskable after a reset.
cfgte	Signal	Slave	Configure exceptions to be taken in thumb mode after a reset.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.

Name	Protocol	Type	Description
dmaexterrirq	Signal	Master	L1 PLE error interrupt.
dmairq	Signal	Master	Interrupt signal from L1 PLE.
dmairq	Signal	Master	Secure interrupt signal from L1 PLE.
fiq	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
irq	Signal	Slave	This signal drives the CPU's interrupt handling.
pmuirq	Signal	Master	Interrupt signal from performance monitoring unit.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
reset	Signal	Slave	Raising this signal will put the core into reset mode.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.
vinithi	Signal	Slave	Configure high vectors after a reset.

## Parameters for ARM Cortex A8CT

### CFGEND0

#### Type

bool

#### Default value

0x0

#### Description

Initialize to BE8 endianness.

### CFGNMFI

#### Type

bool

#### Default value

0x0

#### Description

Enable nonmaskable FIQ interrupts on startup.

### CFGTE

#### Type

bool

#### Default value

0x0

#### Description

Initialize to take exceptions in T32 state. Model starts in T32 state.

**CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`device-accurate-tlb`****Type**

bool

**Default value**

0x0

**Description**

Specify whether all TLBs are modeled.

**implements\_vfp****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has been built with VFP and NEON support.

**l1\_dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l1\_dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l1\_dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l1\_dcache-read\_access\_latency****Type**

int



**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l1\_dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l1\_dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. l1\_dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l1\_dcache-size****Type**

int

**Default value**

0x8000

**Description**

Set L1 D-cache size in bytes.

**l1\_dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Include Level 1 data cache state model.

**l1\_dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l1\_dcachelwrite\_latency is set. This is only used when dcache-state\_modelled=true.

**l1\_dcachelwrite\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. l1\_dcachelwrite\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**l1\_icachelhit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l1\_icachelmiss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l1\_icachelread\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l1_icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`l1_icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `l1_icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l1_icache-size`****Type**

int

**Default value**

0x8000

**Description**

Set L1 I-cache size in bytes.

**`l1_icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Include Level 1 instruction cache state model.

**`l2_cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**l2\_cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l2\_cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l2\_cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2_cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l2\_cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2_cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l2_cache-size`****Type**

int

**Default value**

0x40000

**Description**

Set L2 cache size in bytes.

**`l2_cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l2_cache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Include unified Level 2 cache state model.

**`l2_cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2_cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**l2\_cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2\_cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**master\_id****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**semihosting-ARM\_HLT****Type**

int

**Default value**

0xf000

**Description**

ARM HLT number for semihosting.

**semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

ARM SVC number for semihosting.

**semihosting-Thumb\_HLT****Type**

int

**Default value**

0x3c

**Description**

Thumb HLT number for semihosting.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

Thumb SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**semihosting-hlt-enable****Type**

bool

**Default value**

0x0

**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**semihosting-stack\_base****Type**

int



**Default value**  
0x10000000

**Description**  
Virtual address of base of descending stack.

**semihosting-stack\_limit**

**Type**  
int

**Default value**  
0xf000000

**Description**  
Virtual address of stack limit.

**siliconID**

**Type**  
int

**Default value**  
0x41000000

**Description**  
Value as read by the system coprocessor siliconID register.

**vfp-enable\_at\_reset**

**Type**  
bool

**Default value**  
0x0

**Description**  
Enable coprocessor access and VFP at reset.

3.5.8 ARMCortexA9MPx1CT

ARMCortexA9MPx1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-211: IP revisions support

Revision	Quality level
r3p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About ARMCortexA9MPx1CT

- The following components also exist:
  - ARMCortexA9MPx2CT.
  - ARMCortexA9MPx4CT.

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-3).

- The `ase-present` and `vfp-present` parameters configure the synthesis options:

### **vfp present and ase present**

Neon™ and VFPv3-D32 are supported.

### **vfp present and ase not present**

VFPv3-D16 is supported.

### **vfp not present and ase present**

Illegal. Forces `vfp-present` to true so model has Neon and VFPv3-D32 support.

### **vfp not present and ase not present**

Model has neither Neon nor VFPv3-D32 support.

- If you are using the ARMCortexA9MPx<sub>n</sub>CT component in a VE platform model, the `PERIPHBASE` parameter is set automatically to `0x2C000000` and is not visible in the parameter list.
- When you use the `semihosting-cmd_line` parameter to set the command line that is available to semihosting SVC calls, the value of `argv[0]` is set to the first command-line argument, not to the name of an image.
- Specify `false` for the `device-accurate-tlb` parameter to enable modeling a different number of TLBs if this improves simulation performance. In this case, the simulation is architecturally-accurate, but not device-accurate. Architectural accuracy is almost always sufficient. Specify `true` if device accuracy is required.
- This component implements L1 cache as architecturally defined, but does not implement L2 cache. If you require L2 cache, you can add a PL310 Level 2 Cache Controller component.

## Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- The GIC does not respect the `CFGSDISABLE` signal. This leads to some registers being accessible when they must not be.
- The SCU enable bit is ignored. The SCU is always enabled.
- The SCU ignores the invalidate all register.
- The Broadcast TLB or cache operations in this model do not cause other cores in the cluster that are asleep because of WFI to wake up.
- The RR bit in the SCTLR is ignored.
- The Power Control Register in the system control coprocessor is implemented but writing to it does not change the behavior of the model.
- The model cannot be configured with a 128-entry TLB.

- When modeling the SCU, coherency operations are represented by a memory write followed by a read to refill from memory, rather than using cache-to-cache transfers.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to `false` in production systems.
- The following TLB registers do not have working implementations:
  - Main TLB Attr.
  - Main TLB PA.
  - Main TLB VA.
  - Normal memory remap register.
  - Primary memory remap register.
  - Read Main TLB Entry.
  - Write Main TLB Entry.

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

- Parity error support is hardware-specific so is not modeled.

### Iris and MTI instances for ARMCortexA9MPx1CT

This model has the following Iris instances:

**Table 3-212: ARMCortexA9MPx1CT Iris instances**

InstanceName	ComponentName
ARMCortexA9MPx1CT	Cluster_ARM_Cortex-A9MP
ARMCortexA9MPx1CT.ARM_CortexA9MPx1CT.debug_rom	debug_rom
ARMCortexA9MPx1CT.acp_mapper	PVBusMapper
ARMCortexA9MPx1CT.cpu0	ARM_Cortex-A9MP
ARMCortexA9MPx1CT.cpu0.UTLB	TLB
ARMCortexA9MPx1CT.cpu0.l1dcache	PVCache
ARMCortexA9MPx1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA9MPx1CT.cpu0.l1icache	PVCache
ARMCortexA9MPx1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA9MPx1CT.cpu0.utlb	TLB
ARMCortexA9MPx1CT.ext_bus	PVBusLogger
ARMCortexA9MPx1CT.ext_bus.mapper	PVBusMapper
ARMCortexA9MPx1CT.internal_shareability_remapper	PVBusMapper
ARMCortexA9MPx1CT.l1_incoherent_interconnect	PVCache
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave

InstanceName	ComponentName
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexA9MPx1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-213: ARMCortexA9MPx1CT MTI instances**

InstanceName	ComponentName
ARMCortexA9MPx1CT.acp_mapper	PVBusMapper
ARMCortexA9MPx1CT.cpu0	ARM_Cortex-A9MP
ARMCortexA9MPx1CT.cpu0.UTLB	TLB
ARMCortexA9MPx1CT.cpu0.l1dcache	PVCache
ARMCortexA9MPx1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA9MPx1CT.cpu0.l1icache	PVCache
ARMCortexA9MPx1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA9MPx1CT.ext_bus	PVBusLogger
ARMCortexA9MPx1CT.ext_bus.mapper	PVBusMapper
ARMCortexA9MPx1CT.internal_shareability_remapper	PVBusMapper
ARMCortexA9MPx1CT.l1_incoherent_interconnect	PVCache
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave

InstanceName	ComponentName
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexA9MPx1CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexA9MPx1CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA9MPx1CT

**Table 3-214: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgnmfi[4]	Signal	Slave	This signal disables FIQ mask in CPSR.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
filteren	Signal	Slave	This signal enables filtering of address ranges between master bus ports.
filterend	Value	Slave	This port sets end of region mapped to pvbus_m1.
filterstart	Value	Slave	This port sets start of region mapped to pvbus_m1.
fiq[4]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
fiqout[4]	Signal	Master	This signal exports internal VGIC FIQ signal to the CPU..
ints[224]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
irq[4]	Signal	Slave	This signal drives the CPU's interrupt handling.
irqout[4]	Signal	Master	This signal exports internal VGIC IRQ signal to the CPU.
periphbase	Value	Slave	This port sets the base address of private peripheral region.
periphclk_in	ClockSignal	Slave	The timer and the watchdog take need a clk that is scaled down at least by factor of two.
periphreset	Signal	Slave	This signal resets timer and interrupt controller.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
pvbus_m0	PVBus	Master	AXI master 0 bus master channel.
pvbus_m1	PVBus	Master	AXI master 1 bus master channel.
pwrctl[4]	Value	Slave	This port sets reset value for scu CPU status register.
pwrctl[4]	Value	Master	This port sends scu CPU status register bits.

Name	Protocol	Type	Description
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
scureset	Signal	Slave	This signal resets SCU.
smpnamp[4]	Signal	Master	This signals AMP or SMP mode for each Cortex-A9 processor.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
teinit[4]	Signal	Slave	This signal provides default exception handling state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
wdreset[4]	Signal	Slave	This signal resets individual watchdog.
wdresetreq[4]	Signal	Master	This signal resets rest of the CA9MP system.

## Parameters for ARM Cortex A9 MPx1 CT

### CFGSDISABLE

#### Type

bool

#### Default value

0x0

#### Description

Disable some accesses to GIC registers.

### CLUSTER\_ID

#### Type

int

#### Default value

0x0

#### Description

Processor cluster ID value.

### FILTEREN

#### Type

bool

#### Default value

0x0

#### Description

Enable filtering of accesses through pvbus\_m0.

**FILTEREND****Type**

int

**Default value**

0x0

**Description**

End of region filtered to pvbus\_m0.

**FILTERSTART****Type**

int

**Default value**

0x0

**Description**

Base of region filtered to pvbus\_m0.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Initialize to BE8 endianness.

**cpuX.CFGNMFI****Type**

bool

**Default value**

0x0

**Description**

Enable nonmaskable FIQ interrupts on startup.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.POWERCTLI****Type**

int

**Default value**

0x0

**Description**

Default power control state for processor.

**cpuX.SMPnAMP****Type**

bool

**Default value**

0x0

**Description**

Set whether the processor is part of a coherent domain.



**cpuX.TEINIT****Type**

bool

**Default value**

0x0

**Description**

T32 exception enable. The default has exceptions including reset handled in A32 state.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether model has NEON support.

**cpuX.dcache-size****Type**

int

**Default value**

0x8000

**Description**

Set D-cache size in bytes.

**cpuX.icache-size****Type**

int

**Default value**

0x8000

**Description**

Set I-cache size in bytes.

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-ARM\_HLT****Type**

int

**Default value**

0xf000

**Description**

ARM HLT number for semihosting.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

ARM SVC number for semihosting.

**cpuX.semihosting-Thumb\_HLT****Type**

int

**Default value**

0x3c

**Description**

Thumb HLT number for semihosting.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

Thumb SVC number for semihosting.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-hlt-enable****Type**

bool

**Default value**

0x0

**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable coprocessor access and VFP at reset.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**device-accurate-tlb****Type**

bool

**Default value**

0x0

**Description**

Specify whether all TLBs are modeled.

**dic-spi\_count****Type**

int

**Default value**

0x40

**Description**

Number of shared peripheral interrupts implemented.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

I-cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**`icache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

I-cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

I-cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

I-cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0



Description

I-cache timing annotation latency for read accesses given in ticks per byte accessed.`icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

`icache-state_modelled`

Type

bool

Default value

0x0

Description

Set whether I-cache has stateful implementation.

3.5.9 ARMCortexA9UPCT

ARMCortexA9UPCT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-215: IP revisions support

Revision	Quality level
r3p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexA9UPCT

- The `ase-present` and `vfp-present` parameters configure the synthesis options:
  - vfp present and ase present**  
Neon™ and VFPv3-D32 are supported.
  - vfp present and ase not present**  
VFPv3-D16 is supported.
  - vfp not present and ase present**  
Illegal. Forces `vfp-present` to `true` so model has Neon and VFPv3-D32 support.
  - vfp not present and ase not present**  
Model has neither Neon nor VFPv3-D32 support.
- Specify `false` for the `device-accurate-tlb` parameter to enable modeling a different number of TLBs if this improves simulation performance. In this case, the simulation is architecturally-accurate, but not device-accurate. Architectural accuracy is almost always sufficient. Specify `true` if device accuracy is required.

- When you use the `semihosting-cmd_line` parameter to set the command line that is available to semihosting SVC calls, the value of `argv[0]` is set to the first command-line argument, not to the name of an image.

## Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- The RR bit in the SCTLR is ignored.
- The Power Control Register in the system control coprocessor is implemented but writing to it does not change the behavior of the model.
- The model cannot be configured with a 128-entry TLB.
- This component implements L1 cache as architecturally defined, but does not implement L2 cache. If you require L2 cache you can add a PL310 Level 2 Cache Controller component.
- Parity error support is hardware-specific so is not modeled.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to `false` in production systems.

These TLB registers do not have working implementations:

- Normal memory remap register.
- Primary memory remap register.
- Read Main TLB Entry.
- Write Main TLB Entry.
- Main TLB VA.
- Main TLB PA.
- Main TLB Attr.

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

## Iris and MTI instances for ARMCortexA9UPCT

This model has the following Iris instances:

**Table 3-216: ARMCortexA9UPCT Iris instances**

InstanceName	ComponentName
ARMCortexA9UPCT	Cluster_ARM_Cortex-A9UP
ARMCortexA9UPCT.acp_mapper	PVBusMapper
ARMCortexA9UPCT.cpu0	ARM_Cortex-A9UP
ARMCortexA9UPCT.cpu0.UTLB	TLB
ARMCortexA9UPCT.cpu0.l1dcache	PVCache
ARMCortexA9UPCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA9UPCT.cpu0.l1icache	PVCache

InstanceName	ComponentName
ARMCortexA9UPCT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA9UPCT.cpu0.utlb	TLB
ARMCortexA9UPCT.ext_bus	PVBusLogger
ARMCortexA9UPCT.ext_bus.mapper	PVBusMapper
ARMCortexA9UPCT.l1_incoherent_interconnect	PVCache
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexA9UPCT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-217: ARMCortexA9UPCT MTI instances**

InstanceName	ComponentName
ARMCortexA9UPCT.acp_mapper	PVBusMapper
ARMCortexA9UPCT.cpu0	ARM_Cortex-A9UP
ARMCortexA9UPCT.cpu0.UTLB	TLB
ARMCortexA9UPCT.cpu0.l1dcache	PVCache
ARMCortexA9UPCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA9UPCT.cpu0.l1licache	PVCache
ARMCortexA9UPCT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA9UPCT.ext_bus	PVBusLogger
ARMCortexA9UPCT.ext_bus.mapper	PVBusMapper
ARMCortexA9UPCT.l1_incoherent_interconnect	PVCache
ARMCortexA9UPCT.l1_incoherent_interconnect.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexA9UPCT.11_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexA9UPCT.12_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA9UPCT

**Table 3-218: Ports**

Name	Protocol	Type	Description
cfgend[1]	Signal	Slave	This signal is for EE bit initialisation.
cfgnmfi[1]	Signal	Slave	This signal disables FIQ mask in CPSR.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
cp15sdisable[1]	Signal	Slave	This signal disables write access to some system control processor registers.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[1]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
irq[1]	Signal	Slave	This signal drives the CPU's interrupt handling.
pmuirq[1]	Signal	Master	Interrupt signal from performance monitoring unit.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
reset[1]	Signal	Slave	Raising this signal will put the core into reset mode.
standbywfe[1]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[1]	Signal	Master	This signal indicates if a core is in WFI state.
teinit[1]	Signal	Slave	This signal provides default exception handling state.
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vinithi[1]	Signal	Slave	This signal controls the location of the exception vectors at reset.

## Parameters for ARMCortexA9UPCT

### **CLUSTER\_ID**

#### Type

int

#### Default value

0x0

#### Description

Processor cluster ID value.

### **cpi\_div**

#### Type

int

#### Default value

0x1

#### Description

Divider for calculating CPI (Cycles Per Instruction).

### **cpi\_mul**

#### Type

int

#### Default value

0x1

#### Description

Multiplier for calculating CPI (Cycles Per Instruction).

### **cpuX.CFGEND**

#### Type

bool

#### Default value

0x0

#### Description

Initialize to BE8 endianness.

### **cpuX.CFGNMFI**

#### Type

bool

#### Default value

0x0

**Description**

Enable nonmaskable FIQ interrupts on startup.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.POWERCTLI****Type**

int

**Default value**

0x0

**Description**

Default power control state for processor.

**cpuX.TEINIT****Type**

bool

**Default value**

0x0

**Description**

T32 exception enable. The default has exceptions including reset handled in A32 state.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether model has NEON support.

**cpuX.dcache-size****Type**

int

**Default value**

0x8000

**Description**

Set D-cache size in bytes.

**cpuX.icache-size****Type**

int

**Default value**

0x8000

**Description**

Set I-cache size in bytes.

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-ARM\_HLT****Type**

int

**Default value**

0xf000

**Description**

ARM HLT number for semihosting.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

ARM SVC number for semihosting.

**`cpuX.semihosting-Thumb_HLT`****Type**

int

**Default value**

0x3c

**Description**

Thumb HLT number for semihosting.

**`cpuX.semihosting-Thumb_SVC`****Type**

int

**Default value**

0xab

**Description**

Thumb SVC number for semihosting.

**`cpuX.semihosting-cmd_line`****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**`cpuX.semihosting-cwd`****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**`cpuX.semihosting-enable`****Type**

bool

**Default value**

0x1



**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-hlt-enable****Type**

bool

**Default value**

0x0

**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**`cpuX.semihosting-stack_limit`****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**`cpuX.vfp-enable_at_reset`****Type**

bool

**Default value**

0x0

**Description**

Enable coprocessor access and VFP at reset.

**`cpuX.vfp-present`****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**`dcache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

D-cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**device-accurate-tlb****Type**

bool

**Default value**

0x0

**Description**

Specify whether all TLBs are modeled.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

I-cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**`icache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

I-cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

I-cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

I-cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

I-cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-state_modelled`**

**Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

3.5.10 **ARMCortexA15x1CT**

ARMCortexA15x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-219: IP revisions support

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA15x1CT**

- The following components also exist:
  - ARMCortexA15x2CT.
  - ARMCortexA15x3CT.
  - ARMCortexA15x4CT.

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-3).

- The `ase-present` and `vfp-present` parameters configure the synthesis options:

**vfp present and ase present**

Neon™ and VFPv4-D32 are supported.

**vfp present and ase not present**

VFPv4-D16 is supported.

**vfp not present and ase present**

Illegal. Forces `vfp-present` to `true` so model has Neon and VFPv4-D32 support.

**vfp not present and ase not present**

Model has neither Neon nor VFPv4-D32 support.

- If you are using the `ARMCortexA15x1CT` component on a VE model platform, the `PERIPHBASE` parameter is set automatically to `0x2C000000` and is not visible in the parameter list.

**ACE limitation**

AXI Coherency Extensions (ACE) are extensions to AXI4 that support system-level cache coherency between multiple clusters. The ACE cache models in the Arm® Cortex®-A15 and Cortex-A7, and the ACE support in the CCI-400 have the limitation that they only process one transaction at a time. Normally, the simulation processes each transaction to completion before allowing any master to generate another transaction. However, in the following situation the simulation might fail. If a SystemC bus slave calls `wait()` while it is processing a transaction, this call might allow another master to issue another transaction that passes through the CCI-400 or the Cortex-A15 or Cortex-A7 caches. This situation could happen if a SystemC bus master running in another thread is connected to one of the ACE-lite ports on the CCI-400.

**Differences between the model and the RTL**

This component has the following differences from the corresponding revision of the RTL implementation:

- The GIC does not respect the `CFGSDISABLE` signal. This leads to some registers wrongly being accessible.
- The Broadcast *Translation Lookaside Buffer* (TLB) or cache operations in the model do not cause other cores in the cluster that are asleep because of *Wait For Interrupt* (WFI) to wake up.
- It ignores the RR bit in the SCTLR.
- It implements the Power Control Register in the system control coprocessor but writing to it does not change the behavior of the model.
- When modeling the SCU, coherency operations are by memory writes then reads to refill from memory, rather than cache-to-cache transfers.
- It does not implement ETM registers.
- It implements TLB bitmap registers as RAZ/WI.
- It does not support the Cortex®-A15 mechanism to read the internal memory that the Cache and TLB structures use through the implementation defined region of the system coprocessor interface. This includes the RAM Index Register, IL1DATA Registers, DL1DATA Registers, and associated functionality.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to `false` in production systems.
- ECC and parity schemes are hardware-specific so are not supported.

**Iris and MTI instances for ARMCortexA15x1CT**

This model has the following Iris instances:

**Table 3-220: ARMCortexA15x1CT Iris instances**

InstanceName	ComponentName
ARMCortexA15x1CT	Cluster_ARM_Cortex-A15
ARMCortexA15x1CT.ARM_CortexA15x1CT.debug_rom	debug_rom
ARMCortexA15x1CT.Cortex-A15_GIC	GICv2
ARMCortexA15x1CT.acp_mapper	PVBusMapper
ARMCortexA15x1CT.cpu0	ARM_Cortex-A15
ARMCortexA15x1CT.cpu0.DTLB	TLB
ARMCortexA15x1CT.cpu0.ITLB	TLB
ARMCortexA15x1CT.cpu0.dtlb	TLB
ARMCortexA15x1CT.cpu0.itlb	TLB
ARMCortexA15x1CT.cpu0.l1dcache	PVCache
ARMCortexA15x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA15x1CT.cpu0.l1licache	PVCache
ARMCortexA15x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA15x1CT.ext_bus	PVBusLogger
ARMCortexA15x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA15x1CT.l2_cache	PVCache
ARMCortexA15x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA15x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:



**Table 3-221: ARMCortexA15x1CT MTI instances**

InstanceName	ComponentName
ARMCortexA15x1CT.Cortex-A15_GIC	GICv2
ARMCortexA15x1CT.acp_mapper	PVBusMapper
ARMCortexA15x1CT.cpu0	ARM_Cortex-A15
ARMCortexA15x1CT.cpu0.DTLB	TLB
ARMCortexA15x1CT.cpu0.ITLB	TLB
ARMCortexA15x1CT.cpu0.l1dcache	PVCache
ARMCortexA15x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA15x1CT.cpu0.l1licache	PVCache
ARMCortexA15x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA15x1CT.ext_bus	PVBusLogger
ARMCortexA15x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA15x1CT.l2_cache	PVCache
ARMCortexA15x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA15x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA15x1CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA15x1CT

**Table 3-222: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	Signal	Slave	Enable broadcasting of Inner Shareable transactions.
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.

Name	Protocol	Type	Description
cfgend[4]	Signal	Slave	This signal controls the SCTLR.EE bit.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	This port sets the value in the CLUSTERID field (bits[11:8]) of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Outputs of the generic timers.
CNTPNSIRQ[4]	Signal	Master	Outputs of the generic timers.
CNTPSIRQ[4]	Signal	Master	Outputs of the generic timers.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Outputs of the generic timers.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some secure system control processor registers.
cpuporeset[4]	Signal	Slave	Signal initializes all processor logic including NEON, VFP, Debug, PTM, breakpoint and watchpoint.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
fiqout[4]	Signal	Master	This signal exports internal VGIC FIQ signal to the CPU..
irq[4]	Signal	Slave	This signal drives the CPU's interrupt handling.
irqout[4]	Signal	Master	This signal exports internal VGIC IRQ signal to the CPU.
irqs[224]	Signal	Slave	These signals drive the CPU's interrupt controller interrupt lines.
l2reset	Signal	Slave	This signal resets the shared L2 memory system, interrupt controller and timer logic.
periphbase	Value_64	Slave	This port sets the base address of the private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Signal initializes the shared Debug APB, CTI and CTM logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
standbywfil2	Signal	Master	Indicate that all the individual processors and the L2 memory system are in a WFI state.
teinit[4]	Signal	Slave	This signal enables Thumb exceptions (controls the SCTLR.TE bit).
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vfiq[4]	Signal	Slave	Virtual FIQ inputs. Note that the fiq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
vinithi[4]	Signal	Slave	This signal controls the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ inputs. Note that the irq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).

## Parameters for ARM Cortex-A15x1CT

### **BROADCASTCACHEMAINT**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### **BROADCASTINNER**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

### **BROADCASTOUTER**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

### **CFGSDISABLE**

**Type**

bool

**Default value**

0x0

**Description**

Disable some accesses to GIC registers.

### **CLUSTER\_ID**

**Type**

int

**Default value**

0x0

**Description**

Processor cluster ID value.

**IMINLN****Type**

bool

**Default value**

0x1

**Description**

Instruction cache minimum line size: false=32 bytes, true=64 bytes.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Initialize to BE8 endianness.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.DBGROMADDR****Type**

int

**Default value**

0x12000003

**Description**

This value is used to initialize the CP15 DBGDRAR register. Bits[39:12] of this register specify the ROM table physical address.

**cpuX.DBGROMADDRV****Type**

bool

**Default value**

0x1

**Description**

If true, this sets bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**cpuX.DBGSELFADDR****Type**

int

**Default value**

0x10003

**Description**

This value is used to initialize the CP15 DBGDSAR register. Bits[39:17] of this register specify the ROM table physical address.

**cpuX.DBGSELFADDRV****Type**

bool

**Default value**

0x1

**Description**

If true, this sets bits[1:0] of the CP15 DBGDSAR to indicate that the address is valid.

**cpuX.TEINIT****Type**

bool

**Default value**

0x0

**Description**

T32 exception enable. The default has exceptions including reset handled in A32 state.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether CT model has been built with NEON support.

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-ARM\_HLT****Type**

int

**Default value**

0xf000

**Description**

ARM HLT number for semihosting.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

ARM SVC number for semihosting.

**cpuX.semihosting-Thumb\_HLT****Type**

int

**Default value**

0x3c

**Description**

Thumb HLT number for semihosting.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

Thumb SVC number for semihosting.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-hlt-enable****Type**

bool

**Default value**

0x0



**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable coprocessor access and VFP at reset.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether CT model has been built with VFP support.

**dic-spi\_count****Type**

int

**Default value**

0x40

**Description**

Number of shared peripheral interrupts implemented.

**disable\_periph\_decoder****Type**

bool

**Default value**

0x0

**Description**

Disable memory mapped access to gic system registers.

**internal\_vgic****Type**

bool

**Default value**

0x1

**Description**

Configures whether the model of the processor contains a Virtualized Generic Interrupt Controller (VGIC).

**l1\_dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l1\_dcachel-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l1\_dcachel-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l1\_dcachel-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l1_dcachel-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l1\_dcachel-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `l1_dcachel-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the

time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

### **l1\_dcache-snoop\_data\_transfer\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

### **l1\_dcache-state\_modelled**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Set whether L1 D-cache has stateful implementation.

### **l1\_dcache-write\_access\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l1_dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

### **l1\_dcache-write\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `l1_dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**l1\_icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l1\_icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l1\_icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l1\_icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l1\_icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l1\_icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. l1\_icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l1\_icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether L1 I-cache has stateful implementation.

**l2-data-slice****Type**

int

**Default value**

0x0

**Description**

L2 data RAM slice.

**l2-tag-slice****Type**

int

**Default value**

0x0

**Description**

L2 tag RAM slice.

**l2\_cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**l2\_cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l2\_cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l2\_cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2_cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l2\_cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2_cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l2_cache-size`****Type**

int

**Default value**

0x80000

**Description**

Set L2 cache size in bytes.

**`l2_cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l2_cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l2_cache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether L2 cache has stateful implementation.



**l2\_cache-write\_access\_latency**

Type  
int  
  
Default value  
0x0

Description  
L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2\_cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l2\_cache-write\_latency**

Type  
int  
  
Default value  
0x0

Description  
L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2\_cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**3.5.11 ARMCortexA17x1CT**

ARMCortexA17x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-223: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA17x1CT**

- The following components also exist:
  - ARMCortexA17x2CT.
  - ARMCortexA17x3CT.
  - ARMCortexA17x4CT.

The per-core parameters are preceded by `cpun.`, where *n* identifies the core (0-3).

- The `ase-present` and `vfp-present` parameters configure the synthesis options:

**vfp present and ase present**

NEON and VFPv4-D32 are supported.

**vfp present and ase not present**

VFPv4-D16 is supported.

**vfp not present and ase present**

Illegal. Forces `vfp-present` to `true` so model has NEON and VFPv4-D32 support.

**vfp not present and ase not present**

Model has neither NEON nor VFPv4-D32 support.

- This model exposes the `BROADCASTCACHEMAINT`, `BROADCASTINNER`, and `BROADCASTOUTER` parameters at the CPU level, rather than at the cluster level. To achieve correct behavior, set the same value for all CPUs in the cluster.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to `false` in production systems.
- ECC support is hardware-specific so is not modeled.

**Iris and MTI instances for ARM Cortex A17x1CT**

This model has the following Iris instances:

**Table 3-224: ARM Cortex A17x1CT Iris instances**

InstanceName	ComponentName
ARMCortexA17x1CT	Cluster_ARM_Cortex-A17
ARMCortexA17x1CT.ARM Cortex A17x1CT.debug_rom	debug_rom
ARMCortexA17x1CT.acp_mapper	PVBusMapper
ARMCortexA17x1CT.cpu0	ARM_Cortex-A17
ARMCortexA17x1CT.cpu0.DTLB	TLB
ARMCortexA17x1CT.cpu0.ITLB	TLB
ARMCortexA17x1CT.cpu0.dtlb	TLB
ARMCortexA17x1CT.cpu0.itlb	TLB
ARMCortexA17x1CT.cpu0.l1dcache	PVCache
ARMCortexA17x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA17x1CT.cpu0.l1icache	PVCache
ARMCortexA17x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA17x1CT.ext_bus	PVBusLogger
ARMCortexA17x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA17x1CT.l2_cache	PVCache
ARMCortexA17x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[13]	PVBusSlave

InstanceName	ComponentName
ARMCortexA17x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA17x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-225: ARMCortexA17x1CT MTI instances**

InstanceName	ComponentName
ARMCortexA17x1CT.acp_mapper	PVBusMapper
ARMCortexA17x1CT.cpu0	ARM_Cortex-A17
ARMCortexA17x1CT.cpu0.DTLB	TLB
ARMCortexA17x1CT.cpu0.ITLB	TLB
ARMCortexA17x1CT.cpu0.l1dcache	PVCache
ARMCortexA17x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA17x1CT.cpu0.l1licache	PVCache
ARMCortexA17x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA17x1CT.ext_bus	PVBusLogger
ARMCortexA17x1CT.ext_bus.mapper	PVBusMapper
ARMCortexA17x1CT.l2_cache	PVCache
ARMCortexA17x1CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[3]	PVBusSlave

InstanceName	ComponentName
ARMCortexA17x1CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA17x1CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA17x1CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA17x1CT

**Table 3-226: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	Signal	Slave	Enable broadcasting of Inner Shareable transactions.
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
CFGADDRFILTEENDNS	Value_64	Slave	NS end address (only bits 39:20 are used).
CFGADDRFILTEENDS	Value_64	Slave	S end address (only bits 39:20 are used).
CFGADDRFILTEENNS	Signal	Slave	Enable periph port filtering for NS accesses.
CFGADDRFILTEENS	Signal	Slave	Enable periph port filtering for S accesses.
CFGADDRFILTESTARTNS	Value_64	Slave	NS start address (only bits 39:20 are used).
CFGADDRFILTESTARTS	Value_64	Slave	S start address (only bits 39:20 are used).
cfgend[4]	Signal	Slave	This signal controls the SCTL.R.EE bit.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	This port sets the value in the CLUSTERID field (bits[11:8]) of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Outputs of the generic timers.
CNTPNSIRQ[4]	Signal	Master	Outputs of the generic timers.
CNTPSIRQ[4]	Signal	Master	Outputs of the generic timers.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Outputs of the generic timers.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some secure system control processor registers.
cpuporeset[4]	Signal	Slave	Signal initializes all processor logic including NEON, VFP, Debug, PTM, breakpoint and watchpoint.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
irq[4]	Signal	Slave	This signal drives the CPU's interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete.
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	This signal resets the shared L2 memory system and timer logic.

Name	Protocol	Type	Description
peripheral_m	PVBus	Master	The core's peripheral port. Controlled by filter registers.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Signal initializes the shared Debug APB, CTI and CTM logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
standbywfil2	Signal	Master	Indicate that all the individual processors and the L2 memory system are in a WFI state.
teinit[4]	Signal	Slave	This signal enables Thumb exceptions (controls the SCTLR.TE bit).
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vfiq[4]	Signal	Slave	Virtual FIQ inputs.
vinithi[4]	Signal	Slave	This signal controls the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ inputs.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).

## Parameters for ARM Cortex A17x1CT

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTINNER

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

### BROADCASTOUTER

#### Type

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**CFGADDRFILTENDNS****Type**

int

**Default value**

0x0

**Description**

Peripheral port NS end address.

**CFGADDRFILTENDS****Type**

int

**Default value**

0x0

**Description**

Peripheral port S end address.

**CFGADDRFILTENNS****Type**

bool

**Default value**

0x0

**Description**

Peripheral port NS address filtering enabled.

**CFGADDRFILTENS****Type**

bool

**Default value**

0x0

**Description**

Peripheral port S address filtering enabled.

**CFGADDRFILTSTARTNS****Type**

int

**Default value**

0x0

**Description**

Peripheral port NS start address.

**CFGADDRFILTSTARTS****Type**

int

**Default value**

0x0

**Description**

Peripheral port S start address.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

Processor cluster ID value.

**IMINLN****Type**

bool

**Default value**

0x1

**Description**

Instruction cache minimum line size: false=32 bytes, true=64 bytes.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Initialize to BE8 endianness.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.DBGROMADDR****Type**

int

**Default value**

0x12000003

**Description**

This value is used to initialize the CP15 DBGDRAR register. Bits[39:12] of this register specify the ROM table physical address.

**cpuX.DBGROMADDRV****Type**

bool

**Default value**

0x1



**Description**

If true, this sets bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**cpuX.DBGSELFADDR****Type**

int

**Default value**

0x10003

**Description**

This value is used to initialize the CP15 DBGDSAR register. Bits[39:17] of this register specify the ROM table physical address.

**cpuX.DBGSELFADDRV****Type**

bool

**Default value**

0x1

**Description**

If true, this sets bits[1:0] of the CP15 DBGDSAR to indicate that the address is valid.

**cpuX.TEINIT****Type**

bool

**Default value**

0x0

**Description**

T32 exception enable. The default has exceptions including reset handled in A32 state.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether CT model has been built with NEON support.

**cpuX.l1\_icache-size****Type**

int

**Default value**

0x8000

**Description**

Size of L1 I-cache.

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-ARM\_HLT****Type**

int

**Default value**

0xf000

**Description**

ARM HLT number for semihosting.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

ARM SVC number for semihosting.

**cpuX.semihosting-Thumb\_HLT****Type**

int

**Default value**

0x3c

**Description**

Thumb HLT number for semihosting.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

Thumb SVC number for semihosting.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-hlt-enable****Type**

bool

**Default value**

0x0

**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable coprocessor access and VFP at reset.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether CT model has been built with VFP support.

**l1\_dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l1\_dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l1\_dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l1\_dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l1_dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l1\_dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `l1_dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**l1\_dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l1\_dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether L1 D-cache has stateful implementation.

**l1\_dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l1\_dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l1\_dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. l1\_dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**l1\_icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l1\_icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l1\_icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l1\_icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l1\_icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.



**l1\_icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. l1\_icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l1\_icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether L1 I-cache has stateful implementation.

**l2-data-slice****Type**

int

**Default value**

0x0

**Description**

L2 data RAM slice.

**l2-tag-slice****Type**

int

**Default value**

0x0

**Description**

L2 tag RAM slice.

**l2\_cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**l2\_cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l2\_cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l2\_cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2_cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l2\_cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2_cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l2_cache-size`****Type**

int

**Default value**

0x40000

**Description**

Set L2 cache size in bytes.

**`l2_cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l2_cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l2_cache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether L2 cache has stateful implementation.

**l2\_cache-write\_access\_latency**

Type  
int  
  
Default value  
0x0

Description  
L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2\_cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l2\_cache-write\_latency**

Type  
int  
  
Default value  
0x0

Description  
L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2\_cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**3.5.12 ARMCortexA32CT**

ARMCortexA32CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-227: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA32CT**

- If either L1 cache is stateful, then the L2 cache is stateful. This is controlled by the dcache-state\_modelled and icache-state\_modelled parameters.
- The cache latency parameters are only effective when you enable cache-state modeling.
- Timing annotation for transactions downstream of the cache and TLB models propagates through the models.
- Although Neon support is optional for the Arm® Cortex®-A32 processor, this model does not implement the ase-present parameter. Therefore, it is not possible to configure the model to not support Neon.

- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).
- The `semihosting-cwd` parameter sets the current working directory that is used for semihosting. The host operating system limits the maximum path length. The `semihosting-cwd` parameter does not provide any security. Software running on the model can access files outside this directory using relative paths containing `..` or using absolute paths.
- The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a `PVBusDecoder` to direct traffic to the correct port, `dev_debug_s` or `memorymapped_debug_s`.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. - ECC and parity schemes are hardware-specific so are not supported.

### Iris and MTI instances for ARMCortexA32CT

This model has the following Iris instances:

**Table 3-228: ARMCortexA32CT Iris instances**

InstanceName	ComponentName
ARMCortexA32CT	Cluster_ARM_Cortex-A32
ARMCortexA32CT.AMU	PVBusLogger
ARMCortexA32CT.AMU.mapper	PVBusMapper
ARMCortexA32CT.DAP	PVBusLogger
ARMCortexA32CT.DAP.mapper	PVBusMapper
ARMCortexA32CT.DSU	DSU
ARMCortexA32CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA32CT.MMAP	PVBusLogger
ARMCortexA32CT.MMAP.mapper	PVBusMapper
ARMCortexA32CT.RAS	PVBusLogger
ARMCortexA32CT.RAS.mapper	PVBusMapper
ARMCortexA32CT.acp_mapper	PVBusMapper
ARMCortexA32CT.cpu0	ARM_Cortex-A32
ARMCortexA32CT.cpu0.S1TLB	TLB
ARMCortexA32CT.cpu0.S2TLB	TLB
ARMCortexA32CT.cpu0.UTLB	TLB
ARMCortexA32CT.cpu0.dtlb	TLB
ARMCortexA32CT.cpu0.l1dcache	PVCache
ARMCortexA32CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA32CT.cpu0.l1icache	PVCache
ARMCortexA32CT.cpu0.l1icache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexA32CT.ext_bus	PVBusLogger
ARMCortexA32CT.ext_bus.mapper	PVBusMapper
ARMCortexA32CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA32CT.global_debug_rom	debug_rom
ARMCortexA32CT.l2_cache	PVCache
ARMCortexA32CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA32CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-229: ARMCortexA32CT MTI instances**

InstanceName	ComponentName
ARMCortexA32CT.AMU	PVBusLogger
ARMCortexA32CT.AMU.mapper	PVBusMapper
ARMCortexA32CT.DAP	PVBusLogger
ARMCortexA32CT.DAP.mapper	PVBusMapper
ARMCortexA32CT.DSU	DSU
ARMCortexA32CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA32CT.MMAP	PVBusLogger
ARMCortexA32CT.MMAP.mapper	PVBusMapper
ARMCortexA32CT.RAS	PVBusLogger
ARMCortexA32CT.RAS.mapper	PVBusMapper
ARMCortexA32CT.acp_mapper	PVBusMapper
ARMCortexA32CT.cpu0	ARM_Cortex-A32

InstanceName	ComponentName
ARMCortexA32CT.cpu0.S1TLB	TLB
ARMCortexA32CT.cpu0.S2TLB	TLB
ARMCortexA32CT.cpu0.UTLB	TLB
ARMCortexA32CT.cpu0.l1dcache	PVCache
ARMCortexA32CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA32CT.cpu0.l1icache	PVCache
ARMCortexA32CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA32CT.ext_bus	PVBusLogger
ARMCortexA32CT.ext_bus.mapper	PVBusMapper
ARMCortexA32CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA32CT.l2_cache	PVCache
ARMCortexA32CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA32CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA32CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA32CT

**Table 3-230: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastinner	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
cfgend[4]	Signal	Slave	This signal if for EE bit initialisation.

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
commrx[4]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cp15sdisable2[4]	Signal	Slave	-
cpuporeset[4]	Signal	Slave	CPU power on reset.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross trigger matrix port.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Processor powerup request.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.



Name	Protocol	Type	Description
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete.
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	Level2 reset.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Debug reset.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt
reset[4]	Signal	Slave	Reset.
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
sei[4]	Signal	Slave	Per core System Error physical pins.
smpen[4]	Signal	Master	This signals AMP or SMP mode for each core.
spiden[4]	Signal	Slave	External debug interface.
spniden[4]	Signal	Slave	External debug interface.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state
standbywfil2	Signal	Master	This signal indicated all cores and L2 are idles and in low power state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A32CT

### **BROADCASTCACHEMAINT**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### **BROADCASTINNER**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

### **BROADCASTOUTER**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

### **CLUSTER\_ID**

**Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x22000000

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x1

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.CP15SDISABLE2****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers (FEAT\_CP15SDISABLE2).

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC**

**Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT**

**Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC**

**Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line**

**Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd**

**Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000



**Description**

Virtual address of stack limit.

**`cpuX.trace_special_hlt_imm16`****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

**`cpuX.vfp-enable_at_reset`****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**`cpuX.vfp-present`****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**`dcache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`dcache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`enable_simulation_performance_optimizations`****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

**`icache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**`icache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**`l2cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.



**l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

## Description

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.13 ARMCortexA34CT

ARMCortexA34CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-231: IP revisions support**

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About ARMCortexA34CT

The model has the following features:

- If either L1 cache is stateful, then the L2 cache is stateful. This is controlled by the `dcache-state_modelled` and `icache-state_modelled` parameters.
- The cache latency parameters are only effective when you enable cache-state modeling.
- Timing annotation for transactions downstream of the cache and TLB models propagates through the models.
- Although Neon support is optional for this processor, this model does not implement the `ase-present` parameter. Therefore, it is not possible to configure the model to not support Neon.
- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).
- The `semihosting-cwd` parameter sets the current working directory that is used for semihosting. The host operating system limits the maximum path length. The `semihosting-cwd` parameter does not provide any security. Software running on the model can access files outside this directory using relative paths containing `..` or using absolute paths.
- The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a PVBUSDecoder to direct traffic to the correct port, `dev_debug_s` or `memorymapped_debug_s`.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems.
- This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.
- ECC and parity schemes are hardware-specific so are not supported.

## Iris and MTI instances for ARMCortexA34CT

This model has the following Iris instances:

**Table 3-232: ARMCortexA34CT Iris instances**

InstanceName	ComponentName
ARMCortexA34CT	Cluster_ARM_Cortex-A34
ARMCortexA34CT.AMU	PVBusLogger
ARMCortexA34CT.AMU.mapper	PVBusMapper
ARMCortexA34CT.DAP	PVBusLogger
ARMCortexA34CT.DAP.mapper	PVBusMapper
ARMCortexA34CT.DSU	DSU
ARMCortexA34CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA34CT.MMAP	PVBusLogger
ARMCortexA34CT.MMAP.mapper	PVBusMapper
ARMCortexA34CT.RAS	PVBusLogger
ARMCortexA34CT.RAS.mapper	PVBusMapper
ARMCortexA34CT.acp_mapper	PVBusMapper
ARMCortexA34CT.cpu0	ARM_Cortex-A34
ARMCortexA34CT.cpu0.UTLB	TLB
ARMCortexA34CT.cpu0.dtlb	TLB
ARMCortexA34CT.cpu0.l1dcache	PVCache
ARMCortexA34CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA34CT.cpu0.l1icache	PVCache
ARMCortexA34CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA34CT.ext_bus	PVBusLogger
ARMCortexA34CT.ext_bus.mapper	PVBusMapper
ARMCortexA34CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA34CT.global_debug_rom	debug_rom
ARMCortexA34CT.l2_cache	PVCache
ARMCortexA34CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[6]	PVBusSlave

InstanceName	ComponentName
ARMCortexA34CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA34CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-233: ARMCortexA34CT MTI instances**

InstanceName	ComponentName
ARMCortexA34CT.AMU	PVBusLogger
ARMCortexA34CT.AMU.mapper	PVBusMapper
ARMCortexA34CT.DAP	PVBusLogger
ARMCortexA34CT.DAP.mapper	PVBusMapper
ARMCortexA34CT.DSU	DSU
ARMCortexA34CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA34CT.MMAP	PVBusLogger
ARMCortexA34CT.MMAP.mapper	PVBusMapper
ARMCortexA34CT.RAS	PVBusLogger
ARMCortexA34CT.RAS.mapper	PVBusMapper
ARMCortexA34CT.acp_mapper	PVBusMapper
ARMCortexA34CT.cpu0	ARM_Cortex-A34
ARMCortexA34CT.cpu0.UTLB	TLB
ARMCortexA34CT.cpu0.l1dcache	PVCache
ARMCortexA34CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA34CT.cpu0.l1licache	PVCache
ARMCortexA34CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA34CT.ext_bus	PVBusLogger
ARMCortexA34CT.ext_bus.mapper	PVBusMapper
ARMCortexA34CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA34CT.l2_cache	PVCache
ARMCortexA34CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[2]	PVBusSlave

InstanceName	ComponentName
ARMCortexA34CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA34CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA34CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA34CT

**Table 3-234: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastinner	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
cfgend[4]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
commrx[4]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.

Name	Protocol	Type	Description
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Processor powerup request.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete.
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	This signal resets the shared L2 memory system, interrupt controller and timer logic.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sei[4]	Signal	Slave	Per core virtual System Error physical pins.
smpen[4]	Signal	Master	This signals AMP or SMP mode for each core.

Name	Protocol	Type	Description
spiden[4]	Signal	Slave	External debug interface.
spniden[4]	Signal	Slave	External debug interface.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
standbywfil2	Signal	Master	Indicate that all the individual processors and the L2 memory system are in a WFI state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA34CT

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTINNER

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x0



**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x1

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CFGTE`****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**`cpuX.CP15SDISABLE`****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.CP15SDISABLE2****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers (FEAT\_CP15SDISABLE2).

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0



**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**`dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

### **dcache-write\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

### **enable\_simulation\_performance\_optimizations**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

### **icache-hit\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

### **icache-maintenance\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.14 ARMCortexA35CT

ARMCortexA35CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-235: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About ARMCortexA35CT

The model has the following features:

- If either L1 cache is stateful, then the L2 cache is stateful. This is controlled by the `dcache-state_modelled` and `icache-state_modelled` parameters.
- The cache latency parameters are only effective when you enable cache-state modeling.
- Timing annotation for transactions downstream of the cache and TLB models propagates through the models.
- Although Neon support is optional for this processor, this model does not implement the `ase-present` parameter. Therefore, it is not possible to configure the model to not support Neon.
- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).
- The `semihosting-cwd` parameter sets the current working directory that is used for semihosting. The host operating system limits the maximum path length. The `semihosting-cwd` parameter does not provide any security. Software running on the model can access files outside this directory using relative paths containing `..` or using absolute paths.
- The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a `PVBusDecoder` to direct traffic to the correct port, `dev_debug_s` Of `memorymapped_debug_s`.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems. This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.
- ECC and parity schemes are hardware-specific so are not supported.

## Iris and MTI instances for ARMCortexA35CT

This model has the following Iris instances:

**Table 3-236: ARMCortexA35CT Iris instances**

InstanceName	ComponentName
ARMCortexA35CT	Cluster_ARM_Cortex-A35
ARMCortexA35CT.AMU	PVBusLogger
ARMCortexA35CT.AMU.mapper	PVBusMapper
ARMCortexA35CT.DAP	PVBusLogger
ARMCortexA35CT.DAP.mapper	PVBusMapper
ARMCortexA35CT.DSU	DSU
ARMCortexA35CT.DSU.mpam_busslave	PVBusSlave



InstanceName	ComponentName
ARMCortexA35CT.MMAP	PVBusLogger
ARMCortexA35CT.MMAP.mapper	PVBusMapper
ARMCortexA35CT.RAS	PVBusLogger
ARMCortexA35CT.RAS.mapper	PVBusMapper
ARMCortexA35CT.acp_mapper	PVBusMapper
ARMCortexA35CT.cpu0	ARM_Cortex-A35
ARMCortexA35CT.cpu0.UTLB	TLB
ARMCortexA35CT.cpu0.dtlb	TLB
ARMCortexA35CT.cpu0.l1dcache	PVCache
ARMCortexA35CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA35CT.cpu0.l1licache	PVCache
ARMCortexA35CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA35CT.ext_bus	PVBusLogger
ARMCortexA35CT.ext_bus.mapper	PVBusMapper
ARMCortexA35CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA35CT.global_debug_rom	debug_rom
ARMCortexA35CT.l2_cache	PVCache
ARMCortexA35CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA35CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-237: ARMCortexA35CT MTI instances**

InstanceName	ComponentName
ARMCortexA35CT.AMU	PVBusLogger
ARMCortexA35CT.AMU.mapper	PVBusMapper
ARMCortexA35CT.DAP	PVBusLogger
ARMCortexA35CT.DAP.mapper	PVBusMapper
ARMCortexA35CT.DSU	DSU
ARMCortexA35CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA35CT.MMAP	PVBusLogger
ARMCortexA35CT.MMAP.mapper	PVBusMapper
ARMCortexA35CT.RAS	PVBusLogger
ARMCortexA35CT.RAS.mapper	PVBusMapper
ARMCortexA35CT.acp_mapper	PVBusMapper
ARMCortexA35CT.cpu0	ARM_Cortex-A35
ARMCortexA35CT.cpu0.UTLB	TLB
ARMCortexA35CT.cpu0.l1dcache	PVCache
ARMCortexA35CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA35CT.cpu0.l1icache	PVCache
ARMCortexA35CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA35CT.ext_bus	PVBusLogger
ARMCortexA35CT.ext_bus.mapper	PVBusMapper
ARMCortexA35CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA35CT.l2_cache	PVCache
ARMCortexA35CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA35CT.l2_cache.upstream[9]	PVBusSlave

InstanceName	ComponentName
ARMCortexA35CT.12_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA35CT

**Table 3-238: Ports**

Name	Protocol	Type	Description
aa64naa32[4]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastinner	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
cfgend[4]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
commr[4]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cp15sdisable2[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[4]	Signal	Slave	CPU power on reset.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross trigger matrix port.
dbgack[4]	Signal	Master	External debug interface.

Name	Protocol	Type	Description
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Processor powerup request.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete.
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	Level2 reset.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Debug reset.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt
reset[4]	Signal	Slave	Reset.
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sei[4]	Signal	Slave	Per core System Error physical pins.
smpen[4]	Signal	Master	This signals AMP or SMP mode for each core.
spiden[4]	Signal	Slave	External debug interface.
spniden[4]	Signal	Slave	External debug interface.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state
standbywfi12	Signal	Master	This signal indicated all cores and L2 are idles and in low power state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.

Name	Protocol	Type	Description
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A35CT

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTINNER

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

### BROADCASTOUTER

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x1

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpu_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.AA64nAA32`****Type**

bool

**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CFGTE`****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**`cpuX.CP15SDISABLE`****Type**

bool



**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.CP15SDISABLE2****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers (FEAT\_CP15SDISABLE2).

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**`dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

### **dcache-write\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

### **enable\_simulation\_performance\_optimizations**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

### **icache-hit\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

### **icache-maintenance\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.



**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.15 ARMCortexA53CT

ARMCortexA53CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-239: IP revisions support**

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About ARMCortexA53CT

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a PVBUSDecoder to direct traffic to the correct port, `dev_debug_s` or `memorymapped_debug_s`.

### Differences between the CT model and RTL implementations

This component has the following differences from the corresponding revision of the RTL implementation:

- The value of the AArch64 `PMCEID0_EL0` register, and the AArch32 alias of this register, differs in the model from the TRM value. The model value reflects the model counters.
- The mechanisms for setting the affinity fields of the `MPIDR`. The RTL has two ports:
  - `CLUSTERIDAFF1[7:0]`.
  - `CLUSTERIDAFF2[7:0]`. `AFF1` sets the value of `MPIDR` bits[15:8] and `AFF2` sets the value of `MPIDR` bits[23:16]. In contrast, the model has a single `CLUSTER_ID` port. This difference allows the setting of bits[23:8] of the `MPIDR` using bits[15:0] of the `CLUSTER_ID` value.
- The memory mapped debug registers have a view for cores and a view for external debug agents. In the model, these views require two PVBUS ports. In hardware, the system designer decides how the implementation differentiates the views.
- In the model, a single peer event port combines the functionality of the `eventi` and `evento` signals in the RTL.
- The Generic Timers are Programmer's View (PV) level abstractions: a model-specific protocol connects the `cntvalueb` port to the MemoryMappedCounterModule.
- The GIC CPU Interface is a PV level abstraction: a model-specific protocol connects the GIC CPU Interface to the GIC Distributor.
- The CoreSight Cross Trigger Interface (CTI) is a PV level abstraction: the interface is a model specific one.
- The model has no mechanism to read the internal memory that the Cache and TLB structures use, through the implementation defined region of the system coprocessor interface. This memory includes the RAM Index Register, `IL1DATA` Registers, `DL1DATA` Registers, and associated functionality.
- The model does not implement:
  - ETM registers.
  - The `PMUEVENT` bus.

- The `WARMRESETREQ` signal. However, the warm reset code sequence (see the section [Code sequence to request a Warm reset as a result of `RMR\_ELx.RR` in the \*Arm Architecture Reference Manual for A-profile architecture\*](#) ) makes the model simulate a warm reset of the core.
- The `PMUSNAPSHOTREQ` and `PMUSNAPSHOTACK` signals.
- The `EXTERRIRQ` and `INTERRIRQ` signals.
- Processor dynamic-retention signals.
- The `SYSBARDISABLE` signal.
- This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.
- The `DBGPWRDUP`, `DBGPWRUPREQ`, `DBGNOPWRDWN`, and `DBGIRSTREQ` debug power management signals.
- The RTL synthesis option to remove FP and ASE.
- The RTL synthesis option for a Cortex-A15 style debug memory map.
- Although Neon support is optional for the Cortex-A53 processor, this model does not implement the `ase-present` parameter. This means it is not possible to configure the model to not support Neon.
- ECC and parity schemes are hardware-specific so are not supported.

### Iris and MTI instances for ARMCortexA53CT

This model has the following Iris instances:

**Table 3-240: ARMCortexA53CT Iris instances**

InstanceName	ComponentName
ARMCortexA53CT	Cluster_ARM_Cortex-A53
ARMCortexA53CT.AMU	PVBusLogger
ARMCortexA53CT.AMU.mapper	PVBusMapper
ARMCortexA53CT.DAP	PVBusLogger
ARMCortexA53CT.DAP.mapper	PVBusMapper
ARMCortexA53CT.DSU	DSU
ARMCortexA53CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA53CT.MMAP	PVBusLogger
ARMCortexA53CT.MMAP.mapper	PVBusMapper
ARMCortexA53CT.RAS	PVBusLogger
ARMCortexA53CT.RAS.mapper	PVBusMapper
ARMCortexA53CT.acp_mapper	PVBusMapper
ARMCortexA53CT.cpu0	ARM_Cortex-A53
ARMCortexA53CT.cpu0.UTLB	TLB
ARMCortexA53CT.cpu0.dtlb	TLB
ARMCortexA53CT.cpu0.l1dcache	PVCache

InstanceName	ComponentName
ARMCortexA53CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA53CT.cpu0.l1icache	PVCache
ARMCortexA53CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA53CT.ext_bus	PVBusLogger
ARMCortexA53CT.ext_bus.mapper	PVBusMapper
ARMCortexA53CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA53CT.global_debug_rom	debug_rom
ARMCortexA53CT.l2_cache	PVCache
ARMCortexA53CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA53CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-241: ARMCortexA53CT MTI instances**

InstanceName	ComponentName
ARMCortexA53CT.AMU	PVBusLogger
ARMCortexA53CT.AMU.mapper	PVBusMapper
ARMCortexA53CT.DAP	PVBusLogger
ARMCortexA53CT.DAP.mapper	PVBusMapper
ARMCortexA53CT.DSU	DSU
ARMCortexA53CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA53CT.MMAP	PVBusLogger
ARMCortexA53CT.MMAP.mapper	PVBusMapper
ARMCortexA53CT.RAS	PVBusLogger



InstanceName	ComponentName
ARMCortexA53CT.RAS.mapper	PVBusMapper
ARMCortexA53CT.acp_mapper	PVBusMapper
ARMCortexA53CT.cpu0	ARM_Cortex-A53
ARMCortexA53CT.cpu0.UTLB	TLB
ARMCortexA53CT.cpu0.l1dcache	PVCache
ARMCortexA53CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA53CT.cpu0.l1icache	PVCache
ARMCortexA53CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA53CT.ext_bus	PVBusLogger
ARMCortexA53CT.ext_bus.mapper	PVBusMapper
ARMCortexA53CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
ARMCortexA53CT.l2_cache	PVCache
ARMCortexA53CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA53CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA53CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA53CT

**Table 3-242: Ports**

Name	Protocol	Type	Description
aa64naa32[4]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	Signal	Slave	Enable broadcasting of Inner Shareable transactions.

Name	Protocol	Type	Description
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Timer signals to SoC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SoC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SoC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SoC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
commrx[4]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgll1rstdisable	Signal	Slave	Control ram clear on reset
dbgnopwrdown[4]	Signal	Master	This signals relate to core power down.
dbgpwrupreq[4]	Signal	Master	This signals relate to core power down.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.

Name	Protocol	Type	Description
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete.
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	Reset the shared L2 memory system controller.
l2rstdisable	Signal	Slave	-
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[4]	Signal	Slave	Individual processor RAM Error Interrupt signal input.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sei[4]	Signal	Slave	Per core System Error physical pins.
smpen[4]	Signal	Master	This signals AMP or SMP mode for each core.
spiden[4]	Signal	Slave	External debug interface.
spniden[4]	Signal	Slave	External debug interface.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
standbywfi12	Signal	Master	Indicate that all the individual processors and the L2 systems are in a WFI state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtual FIQ.

Name	Protocol	Type	Description
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Processor Virtual System Error Interrupt request.

## Parameters for ARM Cortex A53CT

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTINNER

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

### BROADCASTOUTER

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

### CLUSTER\_ID

#### Type

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x1

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`**

**Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.AA64nAA32`**

**Type**

bool

**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**`cpuX.CFGEND`**

**Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CFGTE`**

**Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**`cpuX.CP15SDISABLE`**

**Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100



**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.



**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.  
This is only used when dcache-state\_modelled=true.

**l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`l2cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`patch_level`****Type**

int

**Default value**

0x1

**Description**

Patch level of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Revision field in MIDR/MIDR\_EL1. Corresponds to the patch number Y in rXpY.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**revision\_number****Type**

int

**Default value**

0x0

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.16 ARM Cortex A55CT**

ARM Cortex A55CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-243: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARM Cortex A55CT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers’ view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).

- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- Dual ACE masters.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpuN.`, where N identifies the core (0-7).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARMCortexA55CT

This model has the following Iris instances:

**Table 3-244: ARMCortexA55CT Iris instances**

InstanceName	ComponentName
ARMCortexA55CT	Cluster_ARM_Cortex-A55
ARMCortexA55CT.AMU	PVBusLogger
ARMCortexA55CT.AMU.mapper	PVBusMapper
ARMCortexA55CT.DAP	PVBusLogger
ARMCortexA55CT.DAP.mapper	PVBusMapper
ARMCortexA55CT.DSU	DSU
ARMCortexA55CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT.DSU.shared_cache	PVCache
ARMCortexA55CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA55CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA55CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA55CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA55CT.MMAP	PVBusLogger
ARMCortexA55CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT.RAS	PVBusLogger
ARMCortexA55CT.RAS.mapper	PVBusMapper
ARMCortexA55CT.cpu0	ARM_Cortex-A55
ARMCortexA55CT.cpu0.UTLB	TLB
ARMCortexA55CT.cpu0.dtlb	TLB
ARMCortexA55CT.cpu0.l1dcache	PVCache
ARMCortexA55CT.cpu0.l1dcache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexA55CT.cpu0.l1icache	PVCache
ARMCortexA55CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA55CT.cpu0.l2cache	PVCache
ARMCortexA55CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT.ext_bus	PVBusLogger
ARMCortexA55CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT.global_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-245: ARMCortexA55CT MTI instances**

InstanceName	ComponentName
ARMCortexA55CT.AMU	PVBusLogger
ARMCortexA55CT.AMU.mapper	PVBusMapper
ARMCortexA55CT.DAP	PVBusLogger
ARMCortexA55CT.DAP.mapper	PVBusMapper
ARMCortexA55CT.DSU	DSU
ARMCortexA55CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT.DSU.shared_cache	PVCache
ARMCortexA55CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA55CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA55CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA55CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA55CT.MMAP	PVBusLogger
ARMCortexA55CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT.RAS	PVBusLogger
ARMCortexA55CT.RAS.mapper	PVBusMapper
ARMCortexA55CT.cpu0	ARM_Cortex-A55
ARMCortexA55CT.cpu0.UTLB	TLB
ARMCortexA55CT.cpu0.l1dcache	PVCache
ARMCortexA55CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT.cpu0.l1icache	PVCache
ARMCortexA55CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA55CT.cpu0.l2cache	PVCache
ARMCortexA55CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT.ext_bus	PVBusLogger

InstanceName	ComponentName
ARMCortexA55CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA55CT

**Table 3-246: Ports**

Name	Protocol	Type	Description
aa64naa32[8]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[8]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[8]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[8]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[8]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[8]	Signal	Master	Timer signals to SOC.
commirq[8]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[8]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable[8]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[8]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.



Name	Protocol	Type	Description
cryptodisable[8]	Signal	Slave	Disable cryptography extensions after reset.
cti[8]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[8]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[8]	Signal	Master	No power-down request.
dbgpwrupreq[8]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[8]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[8]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[8]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[8]	PChannel	Slave	PChannels for cores
pmuirq[8]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[8]	Signal	Slave	Per core RAM Error Interrupt.
reset[8]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[8]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[8]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.

Name	Protocol	Type	Description
ticks[8]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[8]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[8]	Signal	Slave	Virtualised FIQ.
vinithi[8]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[8]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[8]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A55CT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTOUTER

#### Type

bool

#### Default value

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.AA64nAA32****Type**

bool

**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-size****Type**

int

**Default value**

0x40000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0



**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**`dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

### **dcache-write\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

### **enable\_simulation\_performance\_optimizations**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

### **force\_zero\_PSTATE\_PAN**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

### **has\_acp**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

If true, Accelerator Coherency Port is configured.

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0



**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks.  
This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**  
TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled**

**Type**  
bool

**Default value**  
0x0

**Description**  
If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency**

**Type**  
int

**Default value**  
0x0

**Description**  
Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.17 ARMCortexA55CT\_CortexA75CT

ARMCortexA55CT\_CortexA75CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-247: IP revisions support

Revision	Quality level
CortexA55 r1p0	Full support
CortexA75 r3p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexA55CT\_CortexA75CT

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**  
Possible values are 1-7 (ARMCortexA55CT).

**subcluster1.NUM\_CORES**  
Possible values are 1-4 (ARMCortexA75CT).

The total number of cores in the cluster cannot exceed 8.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- port\_name[0-6] for cores in subcluster0.
- port\_name[7-10] for cores in subcluster1.



All instances in the Master cross trigger matrix port array, `cti[11]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- subcluster0.cpu0 to subcluster0.cpu6 identify cores in subcluster0.
- subcluster1.cpu0 to subcluster1.cpu3 identify cores in subcluster1.

For information about the cores in this model, see:

- [ARMCortexA55CT](#)
- [ARMCortexA75CT](#)

See also [Arm DynamIQ Shared Unit Technical Reference Manual](#).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMCortexA55CT\_CortexA75CT

This model has the following Iris instances:

**Table 3-248: ARMCortexA55CT\_CortexA75CT Iris instances**

InstanceName	ComponentName
ARMCortexA55CT_CortexA75CT	Cluster_ARM_Cortex-A55_Cortex-A75
ARMCortexA55CT_CortexA75CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA75CT.AMU.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.DAP	PVBusLogger
ARMCortexA55CT_CortexA75CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.DSU	DSU
ARMCortexA55CT_CortexA75CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT_CortexA75CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[3]	PVBusSlave

InstanceName	ComponentName
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA55CT_CortexA75CT.MMAP	PVBusLogger
ARMCortexA55CT_CortexA75CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA75CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA75CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA75CT.global_debug_rom	debug_rom
ARMCortexA55CT_CortexA75CT.subcluster0	Subcluster_ARM_Cortex-A55
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.dtlb	TLB
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l1icache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster1	Subcluster_ARM_Cortex-A75
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0	ARM_Cortex-A75
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l1icache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster1.cpu1.dtlb	TLB

This model has the following MTI trace components:

**Table 3-249: ARMCortexA55CT\_CortexA75CT MTI instances**

InstanceName	ComponentName
ARMCortexA55CT_CortexA75CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA75CT.AMU.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.DAP	PVBusLogger



InstanceName	ComponentName
ARMCortexA55CT_CortexA75CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.DSU	DSU
ARMCortexA55CT_CortexA75CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT_CortexA75CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA55CT_CortexA75CT.MMAP	PVBusLogger
ARMCortexA55CT_CortexA75CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA75CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA75CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0	ARM_Cortex-A75
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave

## Ports for ARMCortexA55CT\_CortexA75CT

Table 3-250: Ports

Name	Protocol	Type	Description
aa64naa32[11]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port
AENDMP	Value_64	Slave	DynamlQ port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	DynamlQ port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[11]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[11]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[11]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[11]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[11]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[11]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[11]	Signal	Master	Timer signals to SOC.
commirq[11]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[11]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.

Name	Protocol	Type	Description
cp15sdisable[11]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[11]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[11]	Signal	Slave	Disable cryptography extensions after reset.
cti[11]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[11]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[11]	Signal	Master	No power-down request.
dbgpwrupreq[11]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[11]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[11]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[11]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[11]	PChannel	Slave	PChannels for cores
pmuirq[11]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[11]	Signal	Slave	Per core RAM Error Interrupt.
reset[11]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[11]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[11]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.

Name	Protocol	Type	Description
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[11]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[11]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfirq[11]	Signal	Slave	Virtualised FIQ.
vinithi[11]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[11]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L3 system and becomes coherent (assuming attributes are set correctly).
vsei[11]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA55CT\_CortexA75CT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTOUTER

#### Type

bool

#### Default value

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**`enable_simulation_performance_optimizations`****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**`has_acp`****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**`has_peripheral_port`****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.



**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.AA64nAA32****Type**

bool

**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster0.cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-size****Type**

int

**Default value**

0x40000

**Description**

L2 Cache size in bytes.

**subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.



**subcluster0.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster0.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.

Note: Enabling this parameter will reduce simulation performance.

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.



**subcluster1.CCSIDR-L1D\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

**subcluster1.CCSIDR-L1I\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

**subcluster1.CCSIDR-L2\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul**

**Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.AA64nAA32**

**Type**

bool

**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**subcluster1.cpuX.CFGEND**

**Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CFGTE**

**Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster1.cpuX.CP15SDISABLE**

**Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster1.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`subcluster1.cpuX.l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC**

**Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT**

**Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC**

**Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line**

**Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd**

**Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000



**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`subcluster1.icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster1.invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**`subcluster1.ptw_latency`****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**`subcluster1.tlb_latency`****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled**

**Type**

bool

**Default value**

0x0

**Description**

If true, tlbi invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.18 ARMCortexA55CT\_CortexA76CT**

ARMCortexA55CT\_CortexA76CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-251: IP revisions support**

Revision	Quality level
CortexA55 r1p0	Full support
CortexA76 r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA55CT\_CortexA76CT**

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 1-7 (ARMCortexA55CT).

**subcluster1.NUM\_CORES**

Possible values are 1-4 (ARMCortexA76CT).

The total number of cores in the cluster cannot exceed 8.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- port\_name[0-6] for cores in subcluster0.
- port\_name[7-10] for cores in subcluster1.



All instances in the Master cross trigger matrix port array, `cti[11]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- subcluster0.cpu0 to subcluster0.cpu6 identify cores in subcluster0.
- subcluster1.cpu0 to subcluster1.cpu3 identify cores in subcluster1.

For information about the cores in this model, see:

- [ARMCortexA55CT](#)
- [ARMCortexA76CT](#)

## Iris and MTI instances for ARMCortexA55CT\_CortexA76CT

This model has the following Iris instances:

**Table 3-252: ARMCortexA55CT\_CortexA76CT Iris instances**

InstanceName	ComponentName
ARMCortexA55CT_CortexA76CT	Cluster_ARM_Cortex-A55_Cortex-A76
ARMCortexA55CT_CortexA76CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA76CT.AMU.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.DAP	PVBusLogger
ARMCortexA55CT_CortexA76CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.DSU	DSU
ARMCortexA55CT_CortexA76CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT_CortexA76CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA55CT_CortexA76CT.MMAP	PVBusLogger

InstanceName	ComponentName
ARMCortexA55CT_CortexA76CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA76CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA76CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA76CT.global_debug_rom	debug_rom
ARMCortexA55CT_CortexA76CT.subcluster0	Subcluster_ARM_Cortex-A55
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.dtlb	TLB
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster1	Subcluster_ARM_Cortex-A76
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0	ARM_Cortex-A76
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster1.cpu1.dtlb	TLB

This model has the following MTI trace components:

**Table 3-253: ARMCortexA55CT\_CortexA76CT MTI instances**

InstanceName	ComponentName
ARMCortexA55CT_CortexA76CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA76CT.AMU.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.DAP	PVBusLogger
ARMCortexA55CT_CortexA76CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.DSU	DSU
ARMCortexA55CT_CortexA76CT.DSU.l3_flusher	AsyncCacheFlushUnit



InstanceName	ComponentName
ARMCortexA55CT_CortexA76CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA55CT_CortexA76CT.MMAP	PVBusLogger
ARMCortexA55CT_CortexA76CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA76CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA76CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0	ARM_Cortex-A76
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave

## Ports for ARMCortexA55CT\_CortexA76CT

**Table 3-254: Ports**

Name	Protocol	Type	Description
aa64naa32[11]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port

Name	Protocol	Type	Description
AENDMP	Value_64	Slave	DynamlQ port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	DynamlQ port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[11]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[11]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[11]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[11]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[11]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[11]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[11]	Signal	Master	Timer signals to SOC.
commirq[11]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[11]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable[11]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[11]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[11]	Signal	Slave	Disable cryptography extensions after reset.

Name	Protocol	Type	Description
cti[11]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[11]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[11]	Signal	Master	No power-down request.
dbgpwrupreq[11]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[11]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[11]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[11]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[11]	PChannel	Slave	PChannels for cores
pmbirq[11]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[11]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[11]	Signal	Slave	Per core RAM Error Interrupt.
reset[11]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[11]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[11]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.

Name	Protocol	Type	Description
ticks[11]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[11]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[11]	Signal	Slave	Virtualised FIQ.
vinithi[11]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[11]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L3 system and becomes coherent (assuming attributes are set correctly).
vsei[11]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA55CT\_CortexA76CT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTOUTER

#### Type

bool

#### Default value

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**`enable_simulation_performance_optimizations`****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**`has_acp`****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**`has_peripheral_port`****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.



**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.AA64nAA32****Type**

bool

**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster0.cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-size****Type**

int

**Default value**

0x40000

**Description**

L2 Cache size in bytes.

**subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.



**subcluster0.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster0.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.

Note: Enabling this parameter will reduce simulation performance.

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.



**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster1.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster1.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC**

**Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT**

**Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC**

**Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line**

**Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd**

**Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.



**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**subcluster1.ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**subcluster1.ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**subcluster1.ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster1.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**subcluster1.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.19 ARMCortexA55CT\_CortexA78CT

ARMCortexA55CT\_CortexA78CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-255: IP revisions support**

Revision	Quality level
CortexA55 r1p0	Full support
CortexA78 r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About ARMCortexA55CT\_CortexA78CT

The number of cores in each subcluster is configurable using the following parameters:

#### **subcluster0.NUM\_CORES**

Possible values are 1-7 (ARMCortexA55CT).

#### **subcluster1.NUM\_CORES**

Possible values are 1-4 (ARMCortexA78CT).

The total number of cores in the cluster cannot exceed 8.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- port\_name[0-6] for cores in subcluster0.
- port\_name[7-10] for cores in subcluster1.



All instances in the Master cross trigger matrix port array, `cti[11]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- subcluster0.cpu0 to subcluster0.cpu6 identify cores in subcluster0.
- subcluster1.cpu0 to subcluster1.cpu3 identify cores in subcluster1.

For information about the cores in this model, see:

- [ARMCortexA55CT](#)
- [ARMCortexA78CT](#)

### Iris and MTI instances for ARMCortexA55CT\_CortexA78CT

This model has the following Iris instances:

**Table 3-256: ARMCortexA55CT\_CortexA78CT Iris instances**

InstanceName	ComponentName
ARMCortexA55CT_CortexA78CT	Cluster_ARM_Cortex-A55_Cortex-A78
ARMCortexA55CT_CortexA78CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA78CT.AMU.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA55CT_CortexA78CT.DAP	PVBusLogger
ARMCortexA55CT_CortexA78CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.DSU	DSU
ARMCortexA55CT_CortexA78CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT_CortexA78CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA55CT_CortexA78CT.MMAP	PVBusLogger
ARMCortexA55CT_CortexA78CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA78CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA78CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA78CT.global_debug_rom	debug_rom
ARMCortexA55CT_CortexA78CT.subcluster0	Subcluster_ARM_Cortex-A55
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.dtlb	TLB
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l1icache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1	Subcluster_ARM_Cortex-A78
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0	ARM_Cortex-A78
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1icache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l2cache	PVCache



InstanceName	ComponentName
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu1.dtlb	TLB

This model has the following MTI trace components:

**Table 3-257: ARMCortexA55CT\_CortexA78CT MTI instances**

InstanceName	ComponentName
ARMCortexA55CT_CortexA78CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA78CT.AMU.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.DAP	PVBusLogger
ARMCortexA55CT_CortexA78CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.DSU	DSU
ARMCortexA55CT_CortexA78CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT_CortexA78CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA55CT_CortexA78CT.MMAP	PVBusLogger
ARMCortexA55CT_CortexA78CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA78CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA78CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l1icache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0	ARM_Cortex-A78
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.UTLB	TLB

InstanceName	ComponentName
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1icache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave

## Ports for ARMCortexA55CT\_CortexA78CT

**Table 3-258: Ports**

Name	Protocol	Type	Description
aa64naa32[11]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port
AENDMP	Value_64	Slave	DynamlQ port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	DynamlQ port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[11]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[11]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:14] of the MPIDR.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[11]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[11]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[11]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[11]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[11]	Signal	Master	Timer signals to SOC.
commirq[11]	Signal	Master	Interrupt signal from debug communications channel.

Name	Protocol	Type	Description
core_clk_in[11]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable[11]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[11]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[11]	Signal	Slave	Disable cryptography extensions after reset.
cti[11]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[11]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[11]	Signal	Master	No power-down request.
dbgprupreq[11]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[11]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[11]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[11]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[11]	PChannel	Slave	PChannels for cores
pmbirq[11]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[11]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[11]	Signal	Slave	Per core RAM Error Interrupt.
reset[11]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[11]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain

Name	Protocol	Type	Description
sei[11]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[11]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[11]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[11]	Signal	Slave	Virtualised FIQ.
vinithi[11]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[11]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L3 system and becomes coherent (assuming attributes are set correctly).
vsei[11]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA55CT\_CortexA78CT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are

broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.



**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.AA64nAA32****Type**

bool

**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster0.cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-size****Type**

int

**Default value**

0x40000

**Description**

L2 Cache size in bytes.

**subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.



**subcluster0.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster0.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.

Note: Enabling this parameter will reduce simulation performance.

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.



**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster1.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`subcluster1.cpuX.l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC**

**Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT**

**Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC**

**Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line**

**Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd**

**Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.



**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**subcluster1.ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**subcluster1.ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**subcluster1.ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.has\_coherent\_icache****Type**

bool

**Default value**

0x0

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster1.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster1.icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster1.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat-dcache-cmos-to-pou-as-nop**

**Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**subcluster1.walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.20 ARMCortexA57CT

ARMCortexA57CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-259: IP revisions support

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexA57CT

The per-core parameters are preceded by cpun., where n identifies the core (0-3).

The cache latency parameters are only effective when you enable cache-state modeling. Timing annotation for transactions downstream of the cache and TLB models propagates through the models. This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a PVBUSDecoder to direct traffic to the correct port, dev\_debug\_s or memorymapped\_debug\_s.

## Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- The value of the AArch64 PMCEID0\_ELO register, and the AArch32 alias of this register, differs in the model from the TRM value. The model value reflects the model counters.
- The mechanisms for setting the affinity fields of the MPIDR. The RTL has two ports:
  - CLUSTERIDAFF1[7:0].
  - CLUSTERIDAFF2[7:0]. AFF1 sets the value of MPIDR bits[15:8] and AFF2 sets the value of MPIDR bits[23:16]. In contrast, the model has a single CLUSTER\_ID port. This difference allows the setting of bits[23:8] of the MPIDR using bits[15:0] of the CLUSTER\_ID value.
- The memory mapped debug registers have a view for cores and a view for external debug agents. In the model, these views require two PVBUS ports. In hardware, the system designer decides how the implementation differentiates the views.
- In the model, a single peer event port combines the functionality of the eventi and evento signals in the RTL.
- The Generic Timers are Programmer's View (PV) level abstractions: a model-specific protocol connects the cntvalueb port to the MemoryMappedCounterModule.
- The GIC CPU Interface is a PV level abstraction: a model-specific protocol connects the GIC CPU Interface to the GIC Distributor.
- The CoreSight Cross Trigger Interface (CTI) is a PV level abstraction: the interface is a model-specific one.
- The model has no mechanism to read the internal memory that the Cache and TLB structures use, through the implementation defined region of the system coprocessor interface. This memory includes the RAM Index Register, IL1DATA Registers, DL1DATA Registers, and associated functionality.
- The model does not implement:
  - ETM registers.
  - The PMUEVENT bus.
  - The WARMRESETREQ signal. However, the warm reset code sequence (see the section Code sequence to request a Warm reset as a result of RMR\_ELx.RR in the Arm Architecture Reference Manual for A-profile architecture) makes the model simulate a warm reset of the core.
  - The PMUSNAPSHOTREQ and PMUSNAPSHOTACK signals.
  - The EXTERRIRQ and INTERRIRQ signals.
  - Processor dynamic-retention signals.
  - The SYSBARDISABLE signal.



- This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.
- The `DBGPWRDUP`, `DBGPWRUPREQ`, `DBGNOPWRDWN`, and `DBGIRSTREQ` debug power management signals.
- ECC and parity schemes are hardware-specific so are not supported.

### Iris and MTI instances for ARMCortexA57CT

This model has the following Iris instances:

**Table 3-260: ARMCortexA57CT Iris instances**

InstanceName	ComponentName
ARMCortexA57CT	Cluster_ARM_Cortex-A57
ARMCortexA57CT.AMU	PVBusLogger
ARMCortexA57CT.AMU.mapper	PVBusMapper
ARMCortexA57CT.DAP	PVBusLogger
ARMCortexA57CT.DAP.mapper	PVBusMapper
ARMCortexA57CT.DSU	DSU
ARMCortexA57CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA57CT.MMAP	PVBusLogger
ARMCortexA57CT.MMAP.mapper	PVBusMapper
ARMCortexA57CT.RAS	PVBusLogger
ARMCortexA57CT.RAS.mapper	PVBusMapper
ARMCortexA57CT.acp_mapper	PVBusMapper
ARMCortexA57CT.cpu0	ARM_Cortex-A57
ARMCortexA57CT.cpu0.UTLB	TLB
ARMCortexA57CT.cpu0.dtlb	TLB
ARMCortexA57CT.cpu0.l1dcache	PVCache
ARMCortexA57CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA57CT.cpu0.l1licache	PVCache
ARMCortexA57CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA57CT.ext_bus	PVBusLogger
ARMCortexA57CT.ext_bus.mapper	PVBusMapper
ARMCortexA57CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA57CT.global_debug_rom	debug_rom
ARMCortexA57CT.l2_cache	PVCache
ARMCortexA57CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[13]	PVBusSlave

InstanceName	ComponentName
ARMCortexA57CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA57CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-261: ARMCortexA57CT MTI instances**

InstanceName	ComponentName
ARMCortexA57CT.AMU	PVBusLogger
ARMCortexA57CT.AMU.mapper	PVBusMapper
ARMCortexA57CT.DAP	PVBusLogger
ARMCortexA57CT.DAP.mapper	PVBusMapper
ARMCortexA57CT.DSU	DSU
ARMCortexA57CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA57CT.MMAP	PVBusLogger
ARMCortexA57CT.MMAP.mapper	PVBusMapper
ARMCortexA57CT.RAS	PVBusLogger
ARMCortexA57CT.RAS.mapper	PVBusMapper
ARMCortexA57CT.acp_mapper	PVBusMapper
ARMCortexA57CT.cpu0	ARM_Cortex-A57
ARMCortexA57CT.cpu0.UTLB	TLB
ARMCortexA57CT.cpu0.l1dcache	PVCache
ARMCortexA57CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA57CT.cpu0.l1icache	PVCache
ARMCortexA57CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA57CT.ext_bus	PVBusLogger
ARMCortexA57CT.ext_bus.mapper	PVBusMapper
ARMCortexA57CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA57CT.l2_cache	PVCache
ARMCortexA57CT.l2_cache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexA57CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA57CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA57CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA57CT

**Table 3-262: Ports**

Name	Protocol	Type	Description
aa64naa32[4]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	Signal	Slave	Enable broadcasting of Inner Shareable transactions.
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor

Name	Protocol	Type	Description
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	The per-EL counter signal.
CNTPNSIRQ[4]	Signal	Master	The per-EL counter signal.
CNTPSIRQ[4]	Signal	Master	The per-EL counter signal.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	The per-EL counter signal.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
commrx[4]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	These signals relate to core power down.
dbgpwrupreq[4]	Signal	Master	These signals relate to core power down.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	Reset the shared L2 memory system controller.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.

Name	Protocol	Type	Description
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[4]	Signal	Slave	Individual processor RAM Error Interrupt signal input.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sei[4]	Signal	Slave	Per core System Error physical pins.
smpen[4]	Signal	Master	This signals AMP or SMP mode for each core.
spiden[4]	Signal	Slave	External debug interface.
spniden[4]	Signal	Slave	External debug interface.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
standbywfil2	Signal	Master	Indicate that all the individual processors and the L2 systems are in a WFI state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtual FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Processor Virtual System Error Interrupt request.

## Parameters for ARM Cortex A57CT

### BROADCASTCACHEMAINT

#### Type

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTINNER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x22000000

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x1

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.AA64nAA32****Type**

bool



**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`dcache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit

or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

### **dcache-write\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

### **enable\_simulation\_performance\_optimizations**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

### **icache-hit\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

### **icache-maintenance\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.



**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.  
This is only used when dcache-state\_modelled=true.

**l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`l2cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`ptw_latency`****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency**

**Type**  
int

**Default value**  
0x0

**Description**  
TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled**

**Type**  
bool

**Default value**  
0x0

**Description**  
If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency**

**Type**  
int

**Default value**  
0x0

**Description**  
Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.21 ARMCortexA65AECT

ARMCortexA65AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-263: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexA65AECT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.

- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.
- Split/Lock is supported but with the limitations described in the AE-specific features implemented section.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-7).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.

- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.

As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

## Iris and MTI instances for ARMCortexA65AECT

This model has the following Iris instances:

**Table 3-264: ARMCortexA65AECT Iris instances**

InstanceName	ComponentName
ARMCortexA65AECT	Cluster_ARM_Cortex-A65AE
ARMCortexA65AECT.AMU	PVBusLogger
ARMCortexA65AECT.AMU.mapper	PVBusMapper
ARMCortexA65AECT.DAP	PVBusLogger
ARMCortexA65AECT.DAP.mapper	PVBusMapper
ARMCortexA65AECT.DSU	DSU
ARMCortexA65AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA65AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache	PVCache
ARMCortexA65AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA65AECT.MMAP	PVBusLogger
ARMCortexA65AECT.MMAP.mapper	PVBusMapper
ARMCortexA65AECT.RAS	PVBusLogger
ARMCortexA65AECT.RAS.mapper	PVBusMapper
ARMCortexA65AECT.cpu0.dtlb	TLB
ARMCortexA65AECT.cpu0.thread0	ARM_Cortex-A65AE
ARMCortexA65AECT.cpu0.thread0.UTLB	TLB
ARMCortexA65AECT.cpu0.thread0.l1dcache	PVCache
ARMCortexA65AECT.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu0.thread0.l1icache	PVCache
ARMCortexA65AECT.cpu0.thread0.l1icache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu0.thread0.l2cache	PVCache
ARMCortexA65AECT.cpu0.thread0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu0.thread0.l2cache.upstream[1]	PVBusSlave

InstanceName	ComponentName
ARMCortexA65AECT.cpu0.thread1	ARM_Cortex-A65AE
ARMCortexA65AECT.cpu0.thread1.UTLB	TLB
ARMCortexA65AECT.cpu1.dtlb	TLB
ARMCortexA65AECT.cpu1.thread0	ARM_Cortex-A65AE
ARMCortexA65AECT.cpu1.thread0.UTLB	TLB
ARMCortexA65AECT.cpu1.thread0.l1dcache	PVCache
ARMCortexA65AECT.cpu1.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu1.thread0.l1icache	PVCache
ARMCortexA65AECT.cpu1.thread0.l1icache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu1.thread0.l2cache	PVCache
ARMCortexA65AECT.cpu1.thread0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu1.thread0.l2cache.upstream[1]	PVBusSlave
ARMCortexA65AECT.cpu1.thread1	ARM_Cortex-A65AE
ARMCortexA65AECT.cpu1.thread1.UTLB	TLB
ARMCortexA65AECT.ext_bus	PVBusLogger
ARMCortexA65AECT.ext_bus.mapper	PVBusMapper
ARMCortexA65AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA65AECT.global_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-265: ARMCortexA65AECT MTI instances**

InstanceName	ComponentName
ARMCortexA65AECT.AMU	PVBusLogger
ARMCortexA65AECT.AMU.mapper	PVBusMapper
ARMCortexA65AECT.DAP	PVBusLogger
ARMCortexA65AECT.DAP.mapper	PVBusMapper
ARMCortexA65AECT.DSU	DSU
ARMCortexA65AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA65AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache	PVCache
ARMCortexA65AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA65AECT.MMAP	PVBusLogger
ARMCortexA65AECT.MMAP.mapper	PVBusMapper
ARMCortexA65AECT.RAS	PVBusLogger



InstanceName	ComponentName
ARMCortexA65AECT.RAS.mapper	PVBusMapper
ARMCortexA65AECT.cpu0.thread0	ARM_Cortex-A65AE
ARMCortexA65AECT.cpu0.thread0.UTLB	TLB
ARMCortexA65AECT.cpu0.thread0.l1dcache	PVCache
ARMCortexA65AECT.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu0.thread0.l1icache	PVCache
ARMCortexA65AECT.cpu0.thread0.l1icache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu0.thread0.l2cache	PVCache
ARMCortexA65AECT.cpu0.thread0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu0.thread0.l2cache.upstream[1]	PVBusSlave
ARMCortexA65AECT.cpu0.thread1	ARM_Cortex-A65AE
ARMCortexA65AECT.cpu0.thread1.UTLB	TLB
ARMCortexA65AECT.cpu1.thread0	ARM_Cortex-A65AE
ARMCortexA65AECT.cpu1.thread0.UTLB	TLB
ARMCortexA65AECT.cpu1.thread0.l1dcache	PVCache
ARMCortexA65AECT.cpu1.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu1.thread0.l1icache	PVCache
ARMCortexA65AECT.cpu1.thread0.l1icache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu1.thread0.l2cache	PVCache
ARMCortexA65AECT.cpu1.thread0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT.cpu1.thread0.l2cache.upstream[1]	PVBusSlave
ARMCortexA65AECT.cpu1.thread1	ARM_Cortex-A65AE
ARMCortexA65AECT.cpu1.thread1.UTLB	TLB
ARMCortexA65AECT.ext_bus	PVBusLogger
ARMCortexA65AECT.ext_bus.mapper	PVBusMapper
ARMCortexA65AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA65AECT

**Table 3-266: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[16]	Signal	Slave	This signal is for EE bit initialisation.

Name	Protocol	Type	Description
cfgte[16]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[16]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[16]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[16]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[16]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[16]	Signal	Master	Timer signals to SOC.
commirq[16]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[8]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[8]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[16]	Signal	Slave	Disable cryptography extensions after reset.
cti[16]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[16]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[16]	Signal	Master	These signals relate to core power down.
dbgpwrupreq[16]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[16]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[16]	GICv3Comms	Slave	GICv3 AXI-stream port per thread.

Name	Protocol	Type	Description
irq[16]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[8]	PChannel	Slave	PChannels for cores
pmuirq[16]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[16]	Signal	Slave	Per core RAM Error Interrupt.
reset[8]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[16]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[16]	Signal	Slave	Per core virtual System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[16]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[16]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[16]	Signal	Slave	Virtualised FIQ.
vinithi[16]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[16]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[16]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A65 AECT

### **BROADCASTATOMIC**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### **BROADCASTCACHEMAINT**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### **BROADCASTOUTER**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

### **BROADCASTPERSIST**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x2

**Description**

Number of cores per cluster.

**cluster\_patch\_level****Type**

int

**Default value**

0x0

**Description**

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

**cluster\_revision\_number****Type**

int

**Default value**

0x0

**Description**

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.enable\_single\_thread\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable single thread after reset and keep other thread in reset.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x40000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int



**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.threadY.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.threadY.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.threadY.MPIDR-override****Type**

int

**Default value**

0x0

**Description**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

**cpuX.threadY.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.threadY.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.



**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_lock\_step****Type**

bool

**Default value**

0x0

**Description**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l3cache-size`****Type**

int

**Default value**

0x400000

**Description**

L3 Cache size in bytes.

**`l3cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l3cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l3cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

### **l3cache-write\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

### **pchannel\_treat\_simreset\_as\_poreset**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Register core as ON state to cluster with simulation reset.

### **periph\_address\_end**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

### **periph\_address\_start**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool



**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.22 ARMCortexA65AECT\_CortexA76AECT**

ARMCortexA65AECT\_CortexA76AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-267: IP revisions support**

Revision	Quality level
CortexA65AE r0p0	Full support
CortexA76AE r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA65AECT\_CortexA76AECT**

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 2-6 (ARMCortexA65AECT).

**subcluster1.NUM\_CORES**

Possible values are 2-4 (ARMCortexA76AECT).

The total number of cores in the cluster cannot exceed 8.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- port\_name[0-11] for cores in subcluster0.
  - port\_name[0] is a port for subcluster0.cpu0.thread0
  - port\_name[1] is a port for subcluster0.cpu0.thread1

- port\_name[2] is a port for subcluster0.cpu1.thread0
- port\_name[12-15] for cores in subcluster1.



All instances in the Master cross trigger matrix port array, `cti[16]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- subcluster0.cpu0 to subcluster0.cpu5 identify cores in subcluster0.
- subcluster1.cpu0 to subcluster1.cpu3 identify cores in subcluster1.

For information about the cores in this model, see:

- [ARMCortexA65AECT](#)
- [ARMCortexA76AECT](#)

### Iris and MTI instances for ARMCortexA65AECT\_CortexA76AECT

This model has the following Iris instances:

**Table 3-268: ARMCortexA65AECT\_CortexA76AECT Iris instances**

InstanceName	ComponentName
ARMCortexA65AECT_CortexA76AECT	Cluster_ARM_Cortex-A65AE_Cortex-A76AE
ARMCortexA65AECT_CortexA76AECT.AMU	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.AMU.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.DAP	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.DAP.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.DSU	DSU
ARMCortexA65AECT_CortexA76AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA65AECT_CortexA76AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache	PVCache
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[7]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[8]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[9]	PVBusSlave

InstanceName	ComponentName
ARMCortexA65AECT_CortexA76AECT.MMAP	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.MMAP.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.RAS	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.RAS.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.ext_bus	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.ext_bus.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA65AECT_CortexA76AECT.global_debug_rom	debug_rom
ARMCortexA65AECT_CortexA76AECT.subcluster0	Subcluster_ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.dtlb	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0	ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l1dcache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l1licache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l1licache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l2cache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l2cache.upstream[1]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread1	ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread1.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.dtlb	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0	ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l1dcache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l1licache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l1licache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l2cache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l2cache.upstream[1]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread1	ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread1.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster1	Subcluster_ARM_Cortex-A76AE

InstanceName	ComponentName
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0	ARM_Cortex-A76AE
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1	ARM_Cortex-A76AE
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l1dcache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l1licache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l1licache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l2cache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l2cache.upstream[1]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu2.dtlb	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu3.dtlb	TLB

This model has the following MTI trace components:

**Table 3-269: ARMCortexA65AECT\_CortexA76AECT MTI instances**

InstanceName	ComponentName
ARMCortexA65AECT_CortexA76AECT.AMU	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.AMU.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.DAP	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.DAP.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.DSU	DSU
ARMCortexA65AECT_CortexA76AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA65AECT_CortexA76AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache	PVCache
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[7]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[8]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[9]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.MMAP	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.MMAP.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.RAS	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.RAS.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.ext_bus	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.ext_bus.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.gic_cpuif_decoder_cluster	GlCv3CPUInterfaceDecoder
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0	ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l1dcache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l1icache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l1icache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l2cache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread0.l2cache.upstream[1]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread1	ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu0.thread1.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0	ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l1dcache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l1icache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l1icache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l2cache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread0.l2cache.upstream[1]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread1	ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpu1.thread1.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0	ARM_Cortex-A76AE

InstanceName	ComponentName
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1	ARM_Cortex-A76AE
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.UTLB	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l1dcache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l1licache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l1licache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l2cache	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l2cache.upstream[0]	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpu1.l2cache.upstream[1]	PVBusSlave

## Ports for ARMCortexA65AECT\_CortexA76AECT

**Table 3-270: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port
AENDMP	Value_64	Slave	DynamiQ port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	DynamiQ port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[16]	Signal	Slave	This signal is for EE bit initialisation.
cfgte[16]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[16]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[16]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[16]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[16]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[16]	Signal	Master	Timer signals to SOC.
commirq[16]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[16]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[10]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[16]	Signal	Slave	Disable cryptography extensions after reset.
cti[16]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[16]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[16]	Signal	Master	No power-down request.
dbgpwrupreq[16]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[16]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[16]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[16]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.

Name	Protocol	Type	Description
pchannel_core[10]	PChannel	Slave	PChannels for cores
pmbirq[16]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[16]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[16]	Signal	Slave	Per core RAM Error Interrupt.
reset[10]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[16]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[16]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[16]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[16]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[16]	Signal	Slave	Virtualised FIQ.
vinithi[16]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[16]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L3 system and becomes coherent (assuming attributes are set correctly).
vsei[16]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA65AECT\_CortexA76AECT

### BROADCASTATOMIC

#### Type

bool



**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_lock\_step****Type**

bool

**Default value**

0x0

**Description**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`l3cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`l3cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l3cache-size`****Type**

int

**Default value**

0x100000

**Description**

L3 Cache size in bytes.

**`l3cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l3cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l3cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`l3cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`pchannel_treat_simreset_as_poreset`****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x2

**Description**

Number of cores per cluster.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.enable\_single\_thread\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable single thread after reset and keep other thread in reset.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-maintenance\_latency****Type**

int



**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-size****Type**

int

**Default value**

0x40000

**Description**

L2 Cache size in bytes.

**subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.threadY.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.threadY.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster0.cpuX.threadY.MPIDR-override****Type**

int

**Default value**

0x0

**Description**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

**subcluster0.cpuX.threadY.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.threadY.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.



**subcluster0.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**subcluster0.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster0.icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`subcluster0.invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**`subcluster0.ptw_latency`****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**`subcluster0.reported_patch_level`****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**subcluster0.reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x2

**Description**

Number of cores per cluster.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster1.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_latency`****Type**

int

**Default value**

0x0



**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster1.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC**

**Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT**

**Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC**

**Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line**

**Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd**

**Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**subcluster1.ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**subcluster1.ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**subcluster1.ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.has\_dot\_product****Type**

int

**Default value**

0x2



**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster1.has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**subcluster1.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**subcluster1.reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**subcluster1.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat-dcache-cmos-to-pou-as-nop**

**Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**subcluster1.walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.23 ARMCortexA65CT**

ARMCortexA65CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-271: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA65CT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.

- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGPWRDUP and DBGRSTREQ are not implemented, but DBGPWRUPREQ and DBGNOFWRDWN are implemented.
- Cache stashing capability.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

This model has a variable number of cores per cluster, specified using the NUM\_CORES parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM Cortex-A65CT

This model has the following Iris instances:

**Table 3-272: ARM Cortex-A65CT Iris instances**

InstanceName	ComponentName
ARM Cortex-A65CT	Cluster_ARM_Cortex-A65
ARM Cortex-A65CT.AMU	PVBusLogger
ARM Cortex-A65CT.AMU.mapper	PVBusMapper
ARM Cortex-A65CT.DAP	PVBusLogger

InstanceName	ComponentName
ARMCortexA65CT.DAP.mapper	PVBusMapper
ARMCortexA65CT.DSU	DSU
ARMCortexA65CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA65CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65CT.DSU.shared_cache	PVCache
ARMCortexA65CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA65CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA65CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA65CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA65CT.MMAP	PVBusLogger
ARMCortexA65CT.MMAP.mapper	PVBusMapper
ARMCortexA65CT.RAS	PVBusLogger
ARMCortexA65CT.RAS.mapper	PVBusMapper
ARMCortexA65CT.cpu0.dtlb	TLB
ARMCortexA65CT.cpu0.thread0	ARM_Cortex-A65
ARMCortexA65CT.cpu0.thread0.UTLB	TLB
ARMCortexA65CT.cpu0.thread0.l1dcache	PVCache
ARMCortexA65CT.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65CT.cpu0.thread0.l1licache	PVCache
ARMCortexA65CT.cpu0.thread0.l1licache.upstream[0]	PVBusSlave
ARMCortexA65CT.cpu0.thread0.l2cache	PVCache
ARMCortexA65CT.cpu0.thread0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65CT.cpu0.thread0.l2cache.upstream[1]	PVBusSlave
ARMCortexA65CT.cpu0.thread1	ARM_Cortex-A65
ARMCortexA65CT.cpu0.thread1.UTLB	TLB
ARMCortexA65CT.ext_bus	PVBusLogger
ARMCortexA65CT.ext_bus.mapper	PVBusMapper
ARMCortexA65CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA65CT.global_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-273: ARMCortexA65CT MTI instances**

InstanceName	ComponentName
ARMCortexA65CT.AMU	PVBusLogger
ARMCortexA65CT.AMU.mapper	PVBusMapper
ARMCortexA65CT.DAP	PVBusLogger
ARMCortexA65CT.DAP.mapper	PVBusMapper
ARMCortexA65CT.DSU	DSU
ARMCortexA65CT.DSU.l3_flusher	AsyncCacheFlushUnit

InstanceName	ComponentName
ARMCortexA65CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65CT.DSU.shared_cache	PVCache
ARMCortexA65CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA65CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA65CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA65CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA65CT.MMAP	PVBusLogger
ARMCortexA65CT.MMAP.mapper	PVBusMapper
ARMCortexA65CT.RAS	PVBusLogger
ARMCortexA65CT.RAS.mapper	PVBusMapper
ARMCortexA65CT.cpu0.thread0	ARM_Cortex-A65
ARMCortexA65CT.cpu0.thread0.UTLB	TLB
ARMCortexA65CT.cpu0.thread0.l1dcache	PVCache
ARMCortexA65CT.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65CT.cpu0.thread0.l1licache	PVCache
ARMCortexA65CT.cpu0.thread0.l1licache.upstream[0]	PVBusSlave
ARMCortexA65CT.cpu0.thread0.l2cache	PVCache
ARMCortexA65CT.cpu0.thread0.l2cache.upstream[0]	PVBusSlave
ARMCortexA65CT.cpu0.thread0.l2cache.upstream[1]	PVBusSlave
ARMCortexA65CT.cpu0.thread1	ARM_Cortex-A65
ARMCortexA65CT.cpu0.thread1.UTLB	TLB
ARMCortexA65CT.ext_bus	PVBusLogger
ARMCortexA65CT.ext_bus.mapper	PVBusMapper
ARMCortexA65CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA65CT

**Table 3-274: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[16]	Signal	Slave	This signal is for EE bit initialisation.
cfgte[16]	Signal	Slave	This signal provides default exception handling state.

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[16]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[16]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[16]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[16]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[16]	Signal	Master	Timer signals to SOC.
commirq[16]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[8]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[8]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[16]	Signal	Slave	Disable cryptography extensions after reset.
cti[16]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[16]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[16]	Signal	Master	These signals relate to core power down.
dbgpwrupreq[16]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[16]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[16]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[16]	Signal	Slave	This signal drives the CPUs interrupt handling.



Name	Protocol	Type	Description
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[8]	PChannel	Slave	PChannels for cores
pmuirq[16]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[16]	Signal	Slave	Per core RAM Error Interrupt.
reset[8]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[16]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[16]	Signal	Slave	Per core virtual System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[16]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[16]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[16]	Signal	Slave	Virtualised FIQ.
vinithi[16]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[16]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[16]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA65CT

### BROADCASTATOMIC

#### Type

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**cluster\_patch\_level****Type**

int

**Default value**

0x0

**Description**

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

**cluster\_revision\_number****Type**

int

**Default value**

0x0

**Description**

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.enable\_single\_thread\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable single thread after reset and keep other thread in reset.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x40000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.



**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.threadY.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.threadY.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.threadY.MPIDR-override****Type**

int

**Default value**

0x0

**Description**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

**cpuX.threadY.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.threadY.VINITI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.



**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**l3cache-size****Type**

int

**Default value**

0x400000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled**

**Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.24 ARM CortexA72CT**

ARM CortexA72CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-275: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARM CortexA72CT**

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

- The cache latency cluster parameters are only effective when you enable cache-state modeling.
- Timing annotation for transactions downstream of the cache and TLB models propagates through the cache and TLB models.
- ECC and parity schemes are hardware-specific so are not supported.
- This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM Cortex-A72CT

This model has the following Iris instances:

**Table 3-276: ARM Cortex-A72CT Iris instances**

InstanceName	ComponentName
ARM Cortex-A72CT	Cluster_ARM_Cortex-A72
ARM Cortex-A72CT.AMU	PVBusLogger
ARM Cortex-A72CT.AMU.mapper	PVBusMapper
ARM Cortex-A72CT.DAP	PVBusLogger
ARM Cortex-A72CT.DAP.mapper	PVBusMapper
ARM Cortex-A72CT.DSU	DSU
ARM Cortex-A72CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex-A72CT.MMAP	PVBusLogger
ARM Cortex-A72CT.MMAP.mapper	PVBusMapper
ARM Cortex-A72CT.RAS	PVBusLogger
ARM Cortex-A72CT.RAS.mapper	PVBusMapper
ARM Cortex-A72CT.acp_mapper	PVBusMapper
ARM Cortex-A72CT.cpu0	ARM_Cortex-A72
ARM Cortex-A72CT.cpu0.UTLB	TLB
ARM Cortex-A72CT.cpu0.dtlb	TLB
ARM Cortex-A72CT.cpu0.l1dcache	PVCache
ARM Cortex-A72CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM Cortex-A72CT.cpu0.l1licache	PVCache
ARM Cortex-A72CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARM Cortex-A72CT.ext_bus	PVBusLogger
ARM Cortex-A72CT.ext_bus.mapper	PVBusMapper
ARM Cortex-A72CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARM Cortex-A72CT.global_debug_rom	debug_rom
ARM Cortex-A72CT.l2_cache	PVCache
ARM Cortex-A72CT.l2_cache.upstream[0]	PVBusSlave
ARM Cortex-A72CT.l2_cache.upstream[10]	PVBusSlave
ARM Cortex-A72CT.l2_cache.upstream[11]	PVBusSlave
ARM Cortex-A72CT.l2_cache.upstream[12]	PVBusSlave
ARM Cortex-A72CT.l2_cache.upstream[13]	PVBusSlave
ARM Cortex-A72CT.l2_cache.upstream[14]	PVBusSlave
ARM Cortex-A72CT.l2_cache.upstream[15]	PVBusSlave
ARM Cortex-A72CT.l2_cache.upstream[16]	PVBusSlave

InstanceName	ComponentName
ARMCortexA72CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA72CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-277: ARMCortexA72CT MTI instances**

InstanceName	ComponentName
ARMCortexA72CT.AMU	PVBusLogger
ARMCortexA72CT.AMU.mapper	PVBusMapper
ARMCortexA72CT.DAP	PVBusLogger
ARMCortexA72CT.DAP.mapper	PVBusMapper
ARMCortexA72CT.DSU	DSU
ARMCortexA72CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA72CT.MMAP	PVBusLogger
ARMCortexA72CT.MMAP.mapper	PVBusMapper
ARMCortexA72CT.RAS	PVBusLogger
ARMCortexA72CT.RAS.mapper	PVBusMapper
ARMCortexA72CT.acp_mapper	PVBusMapper
ARMCortexA72CT.cpu0	ARM_Cortex-A72
ARMCortexA72CT.cpu0.UTLB	TLB
ARMCortexA72CT.cpu0.l1dcache	PVCache
ARMCortexA72CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA72CT.cpu0.l1icache	PVCache
ARMCortexA72CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA72CT.ext_bus	PVBusLogger
ARMCortexA72CT.ext_bus.mapper	PVBusMapper
ARMCortexA72CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA72CT.l2_cache	PVCache
ARMCortexA72CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[12]	PVBusSlave



InstanceName	ComponentName
ARMCortexA72CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA72CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA72CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA72CT

**Table 3-278: Ports**

Name	Protocol	Type	Description
aa64naa32[4]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	Signal	Slave	Enable broadcasting of Inner Shareable transactions.
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	The per-EL counter signal.
CNTPNSIRQ[4]	Signal	Master	The per-EL counter signal.

Name	Protocol	Type	Description
CNTPSIRQ[4]	Signal	Master	The per-EL counter signal.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module
CNTVIRQ[4]	Signal	Master	The per-EL counter signal.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
commrx[4]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	These signals relate to core power down.
dbgpwrupreq[4]	Signal	Master	These signals relate to core power down.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete.
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	Reset the shared L2 memory system controller.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.

Name	Protocol	Type	Description
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[4]	Signal	Slave	Individual processor RAM Error Interrupt signal input.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sei[4]	Signal	Slave	Per core System Error physical pins.
smpen[4]	Signal	Master	This signals AMP or SMP mode for each core.
spiden[4]	Signal	Slave	External debug interface.
spniden[4]	Signal	Slave	External debug interface.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
standbywfil2	Signal	Master	Indicate that all the individual processors and the L2 systems are in a WFI state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtual FIQ
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Processor Virtual System Error Interrupt request.

## Parameters for ARM Cortex A72CT

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTINNER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are

broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x22000000

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x1

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.AA64nAA32****Type**

bool

**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.



**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`dcache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`enable_simulation_performance_optimizations`****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

**`icache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**`icache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**`l2cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks.  
This is only used when `dcache-state_modelled=true`.

**`l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.



**l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled**

**Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.25 ARMCortexA73CT**

ARMCortexA73CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-279: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA73CT**

- The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).
- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).
- If either L1 cache is stateful, then the L2 cache is stateful. This is controlled by the `dcache-state_modelled` and `icache-state_modelled` parameters.

- The `semihosting-cwd` parameter sets the CWD that is used for semihosting. The host operating system limits the maximum path length. The `semihosting-cwd` parameter does not provide any security. Software running on the model can access files outside this directory using relative paths containing `..` or using absolute paths.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

- The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a `PVBusDecoder` to direct traffic to the correct port, `dev_debug_s` Of `memorymapped_debug_s`.
- ECC support is hardware-specific so is not modeled.

## Iris and MTI instances for ARMCortexA73CT

This model has the following Iris instances:

**Table 3-280: ARMCortexA73CT Iris instances**

InstanceName	ComponentName
ARMCortexA73CT	Cluster_ARM_Cortex-A73
ARMCortexA73CT.AMU	PVBusLogger
ARMCortexA73CT.AMU.mapper	PVBusMapper
ARMCortexA73CT.DAP	PVBusLogger
ARMCortexA73CT.DAP.mapper	PVBusMapper
ARMCortexA73CT.DSU	DSU
ARMCortexA73CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA73CT.MMAP	PVBusLogger
ARMCortexA73CT.MMAP.mapper	PVBusMapper
ARMCortexA73CT.RAS	PVBusLogger
ARMCortexA73CT.RAS.mapper	PVBusMapper
ARMCortexA73CT.acp_mapper	PVBusMapper
ARMCortexA73CT.cpu0	ARM_Cortex-A73
ARMCortexA73CT.cpu0.UTLB	TLB
ARMCortexA73CT.cpu0.dtlb	TLB
ARMCortexA73CT.cpu0.l1dcache	PVCache
ARMCortexA73CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA73CT.cpu0.l1icache	PVCache
ARMCortexA73CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA73CT.ext_bus	PVBusLogger
ARMCortexA73CT.ext_bus.mapper	PVBusMapper
ARMCortexA73CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA73CT.global_debug_rom	debug_rom
ARMCortexA73CT.l2_cache	PVCache
ARMCortexA73CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[10]	PVBusSlave

InstanceName	ComponentName
ARMCortexA73CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA73CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-281: ARMCortexA73CT MTI instances**

InstanceName	ComponentName
ARMCortexA73CT.AMU	PVBusLogger
ARMCortexA73CT.AMU.mapper	PVBusMapper
ARMCortexA73CT.DAP	PVBusLogger
ARMCortexA73CT.DAP.mapper	PVBusMapper
ARMCortexA73CT.DSU	DSU
ARMCortexA73CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA73CT.MMAP	PVBusLogger
ARMCortexA73CT.MMAP.mapper	PVBusMapper
ARMCortexA73CT.RAS	PVBusLogger
ARMCortexA73CT.RAS.mapper	PVBusMapper
ARMCortexA73CT.acp_mapper	PVBusMapper
ARMCortexA73CT.cpu0	ARM_Cortex-A73
ARMCortexA73CT.cpu0.UTLB	TLB
ARMCortexA73CT.cpu0.l1dcache	PVCache
ARMCortexA73CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA73CT.cpu0.l1icache	PVCache
ARMCortexA73CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA73CT.ext_bus	PVBusLogger
ARMCortexA73CT.ext_bus.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA73CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA73CT.l2_cache	PVCache
ARMCortexA73CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA73CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA73CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA73CT

**Table 3-282: Ports**

Name	Protocol	Type	Description
aa64naa32[4]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastinner	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
cfgend[4]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor

Name	Protocol	Type	Description
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
commrx[4]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[4]	Signal	Slave	CPU power on reset.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross trigger matrix port.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Processor powerup request.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2flushdone	Signal	Master	Flush of L2 memory system complete.
l2flushreq	Signal	Slave	Request flush of L2 memory system.
l2reset	Signal	Slave	Level2 reset.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Debug reset.

Name	Protocol	Type	Description
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt
reset[4]	Signal	Slave	Reset.
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sei[4]	Signal	Slave	Per core System Error physical pins.
smpen[4]	Signal	Master	This signals AMP or SMP mode for each core.
spiden[4]	Signal	Slave	External debug interface.
spniden[4]	Signal	Slave	External debug interface.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state
standbywfil2	Signal	Master	This signal indicated all cores and L2 are idles and in low power state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A73CT

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.



**BROADCASTINNER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**CCSIDR-L1D\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L1I\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L2\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x1

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`**

**Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.AA64nAA32`**

**Type**

bool

**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**`cpuX.CFGEND`**

**Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CFGTE`**

**Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**`cpuX.CP15SDISABLE`**

**Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.SMPEN****Type**

bool

**Default value**

0x0

**Description**

Enable broadcast messages necessary for correct SMP operation at reset.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int



**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**revision\_number****Type**

int

**Default value**

0x0



**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**tlb\_latency**

**Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled**

**Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.26 ARMCortexA75CT

ARMCortexA75CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-283: IP revisions support

Revision	Quality level
r3p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About ARMCortexA75CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGWRUPREQ and DBGWRUPREQ are not implemented, but DBGWRUPREQ and DBGWRUPREQ are implemented.
- Cache stashing capability.
- Dual ACE masters.
- This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).
- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMCortexA75CT

This model has the following Iris instances:

**Table 3-284: ARMCortexA75CT Iris instances**

InstanceName	ComponentName
ARMCortexA75CT	Cluster_ARM_Cortex-A75
ARMCortexA75CT.AMU	PVBusLogger
ARMCortexA75CT.AMU.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA75CT.DAP	PVBusLogger
ARMCortexA75CT.DAP.mapper	PVBusMapper
ARMCortexA75CT.DSU	DSU
ARMCortexA75CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA75CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA75CT.DSU.shared_cache	PVCache
ARMCortexA75CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA75CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA75CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA75CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA75CT.MMAP	PVBusLogger
ARMCortexA75CT.MMAP.mapper	PVBusMapper
ARMCortexA75CT.RAS	PVBusLogger
ARMCortexA75CT.RAS.mapper	PVBusMapper
ARMCortexA75CT.cpu0	ARM_Cortex-A75
ARMCortexA75CT.cpu0.UTLB	TLB
ARMCortexA75CT.cpu0.dtlb	TLB
ARMCortexA75CT.cpu0.l1dcache	PVCache
ARMCortexA75CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA75CT.cpu0.l1icache	PVCache
ARMCortexA75CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA75CT.cpu0.l2cache	PVCache
ARMCortexA75CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA75CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA75CT.ext_bus	PVBusLogger
ARMCortexA75CT.ext_bus.mapper	PVBusMapper
ARMCortexA75CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA75CT.global_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-285: ARMCortexA75CT MTI instances**

InstanceName	ComponentName
ARMCortexA75CT.AMU	PVBusLogger
ARMCortexA75CT.AMU.mapper	PVBusMapper
ARMCortexA75CT.DAP	PVBusLogger
ARMCortexA75CT.DAP.mapper	PVBusMapper
ARMCortexA75CT.DSU	DSU
ARMCortexA75CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA75CT.DSU.mpam_busslave	PVBusSlave

InstanceName	ComponentName
ARMCortexA75CT.DSU.shared_cache	PVCache
ARMCortexA75CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA75CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA75CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA75CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA75CT.MMAP	PVBusLogger
ARMCortexA75CT.MMAP.mapper	PVBusMapper
ARMCortexA75CT.RAS	PVBusLogger
ARMCortexA75CT.RAS.mapper	PVBusMapper
ARMCortexA75CT.cpu0	ARM_Cortex-A75
ARMCortexA75CT.cpu0.UTLB	TLB
ARMCortexA75CT.cpu0.l1dcache	PVCache
ARMCortexA75CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA75CT.cpu0.l1icache	PVCache
ARMCortexA75CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA75CT.cpu0.l2cache	PVCache
ARMCortexA75CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA75CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA75CT.ext_bus	PVBusLogger
ARMCortexA75CT.ext_bus.mapper	PVBusMapper
ARMCortexA75CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA75CT

**Table 3-286: Ports**

Name	Protocol	Type	Description
aa64naa32[4]	Signal	Slave	Register width after reset.
acp_s	PVBus	Slave	AXI ACP slave port
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[4]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[4]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[4]	PChannel	Slave	PChannels for cores

Name	Protocol	Type	Description
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Individual processor RAM Error Interrupt signal input.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core virtual System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtual FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtual IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A75CT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CCSIDR-L1D\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L1I\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L2\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are



broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.AA64nAA32****Type**

bool

**Default value**

0x1

**Description**

Register width configuration at reset. 0, AArch32. 1, AArch64.

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**`cpuX.semihosting-T32_HLT`****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**`cpuX.semihosting-Thumb_SVC`****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**`cpuX.semihosting-cmd_line`****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**`cpuX.semihosting-cwd`****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**`cpuX.semihosting-enable`****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000



**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit

or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

### **l3cache-read\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

### **l3cache-size**

#### **Type**

int

#### **Default value**

0x80000

#### **Description**

L3 Cache size in bytes.

### **l3cache-snoop\_data\_transfer\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

### **l3cache-snoop\_issue\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.



**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.27 ARMCortexA76AECT**

ARMCortexA76AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-287: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA76AECT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGPWRDUP and DBGIRSTREQ are not implemented, but DBGPWRUPREQ and DBGNOFWRDWN are implemented.
- Cache stashing capability.

- Split/Lock is supported but with the limitations described in the AE-specific features implemented section.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpu`., where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.
- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.

As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

## Iris and MTI instances for ARM CortexA76AECT

This model has the following Iris instances:

**Table 3-288: ARM CortexA76AECT Iris instances**

InstanceName	ComponentName
ARM CortexA76AECT	Cluster_ARM_Cortex-A76AE
ARM CortexA76AECT.AMU	PVBusLogger
ARM CortexA76AECT.AMU.mapper	PVBusMapper
ARM CortexA76AECT.DAP	PVBusLogger
ARM CortexA76AECT.DAP.mapper	PVBusMapper
ARM CortexA76AECT.DSU	DSU
ARM CortexA76AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM CortexA76AECT.DSU.mpam_busslave	PVBusSlave
ARM CortexA76AECT.DSU.shared_cache	PVCache
ARM CortexA76AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM CortexA76AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM CortexA76AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM CortexA76AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARM CortexA76AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARM CortexA76AECT.DSU.shared_cache.upstream[5]	PVBusSlave

InstanceName	ComponentName
ARMCortexA76AECT.MMAP	PVBusLogger
ARMCortexA76AECT.MMAP.mapper	PVBusMapper
ARMCortexA76AECT.RAS	PVBusLogger
ARMCortexA76AECT.RAS.mapper	PVBusMapper
ARMCortexA76AECT.cpu0	ARM_Cortex-A76AE
ARMCortexA76AECT.cpu0.UTLB	TLB
ARMCortexA76AECT.cpu0.debug_rom	debug_rom
ARMCortexA76AECT.cpu0.dtlb	TLB
ARMCortexA76AECT.cpu0.l1dcache	PVCache
ARMCortexA76AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu0.l1licache	PVCache
ARMCortexA76AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu0.l2cache	PVCache
ARMCortexA76AECT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA76AECT.cpu1	ARM_Cortex-A76AE
ARMCortexA76AECT.cpu1.UTLB	TLB
ARMCortexA76AECT.cpu1.debug_rom	debug_rom
ARMCortexA76AECT.cpu1.dtlb	TLB
ARMCortexA76AECT.cpu1.l1dcache	PVCache
ARMCortexA76AECT.cpu1.l1dcache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu1.l1licache	PVCache
ARMCortexA76AECT.cpu1.l1licache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu1.l2cache	PVCache
ARMCortexA76AECT.cpu1.l2cache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu1.l2cache.upstream[1]	PVBusSlave
ARMCortexA76AECT.ext_bus	PVBusLogger
ARMCortexA76AECT.ext_bus.mapper	PVBusMapper
ARMCortexA76AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA76AECT.global_debug_rom	debug_rom
ARMCortexA76AECT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-289: ARMCortexA76AECT MTI instances**

InstanceName	ComponentName
ARMCortexA76AECT.AMU	PVBusLogger
ARMCortexA76AECT.AMU.mapper	PVBusMapper
ARMCortexA76AECT.DAP	PVBusLogger
ARMCortexA76AECT.DAP.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA76AECT.DSU	DSU
ARMCortexA76AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA76AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA76AECT.DSU.shared_cache	PVCache
ARMCortexA76AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA76AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA76AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA76AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA76AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA76AECT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA76AECT.MMAP	PVBusLogger
ARMCortexA76AECT.MMAP.mapper	PVBusMapper
ARMCortexA76AECT.RAS	PVBusLogger
ARMCortexA76AECT.RAS.mapper	PVBusMapper
ARMCortexA76AECT.cpu0	ARM_Cortex-A76AE
ARMCortexA76AECT.cpu0.UTLB	TLB
ARMCortexA76AECT.cpu0.l1dcache	PVCache
ARMCortexA76AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu0.l1licache	PVCache
ARMCortexA76AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu0.l2cache	PVCache
ARMCortexA76AECT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA76AECT.cpu1	ARM_Cortex-A76AE
ARMCortexA76AECT.cpu1.UTLB	TLB
ARMCortexA76AECT.cpu1.l1dcache	PVCache
ARMCortexA76AECT.cpu1.l1dcache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu1.l1licache	PVCache
ARMCortexA76AECT.cpu1.l1licache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu1.l2cache	PVCache
ARMCortexA76AECT.cpu1.l2cache.upstream[0]	PVBusSlave
ARMCortexA76AECT.cpu1.l2cache.upstream[1]	PVBusSlave
ARMCortexA76AECT.ext_bus	PVBusLogger
ARMCortexA76AECT.ext_bus.mapper	PVBusMapper
ARMCortexA76AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA76AECT

Table 3-290: Ports

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[4]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.

Name	Protocol	Type	Description
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	These signals relate to core power down.
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core virtual System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.



Name	Protocol	Type	Description
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A76 AECT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTOUTER

#### Type

bool

#### Default value

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x2

**Description**

Number of cores per cluster.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTL.R.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter `trace_special_hlt_imm16`.

**`cpuX.l2cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**`cpuX.semihosting-A64_HLT`****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**`cpuX.semihosting-ARM_SVC`****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**`cpuX.semihosting-T32_HLT`****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**`cpuX.semihosting-Thumb_SVC`****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**`cpuX.semihosting-cmd_line`****Type**

string

**Default value**

""



**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**`cpuX.semihosting-stack_limit`****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**`cpuX.trace_special_hlt_imm16`****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**`cpuX.vfp-enable_at_reset`****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**`dcache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_lock\_step****Type**

bool

**Default value**

0x0

**Description**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.



**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x100000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled**

**Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat-dcache-cmos-to-pou-as-nop**

**Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.28 ARMCortexA76CT**

ARMCortexA76CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-291: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA76CT**

The model supports the following features:

- DynamlQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamlQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGWRUP and DBGRSTREQ are not implemented, but DBGWRUPREQ and DBGNOFWRDWN are implemented.
- Cache stashing capability.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMCortexA76CT

This model has the following Iris instances:

**Table 3-292: ARMCortexA76CT Iris instances**

InstanceName	ComponentName
ARMCortexA76CT	Cluster_ARM_Cortex-A76
ARMCortexA76CT.AMU	PVBusLogger
ARMCortexA76CT.AMU.mapper	PVBusMapper
ARMCortexA76CT.DAP	PVBusLogger

InstanceName	ComponentName
ARMCortexA76CT.DAP.mapper	PVBusMapper
ARMCortexA76CT.DSU	DSU
ARMCortexA76CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA76CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA76CT.DSU.shared_cache	PVCache
ARMCortexA76CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA76CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA76CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA76CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA76CT.MMAP	PVBusLogger
ARMCortexA76CT.MMAP.mapper	PVBusMapper
ARMCortexA76CT.RAS	PVBusLogger
ARMCortexA76CT.RAS.mapper	PVBusMapper
ARMCortexA76CT.cpu0	ARM_Cortex-A76
ARMCortexA76CT.cpu0.UTLB	TLB
ARMCortexA76CT.cpu0.debug_rom	debug_rom
ARMCortexA76CT.cpu0.dtlb	TLB
ARMCortexA76CT.cpu0.l1dcache	PVCache
ARMCortexA76CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA76CT.cpu0.l1licache	PVCache
ARMCortexA76CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA76CT.cpu0.l2cache	PVCache
ARMCortexA76CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA76CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA76CT.ext_bus	PVBusLogger
ARMCortexA76CT.ext_bus.mapper	PVBusMapper
ARMCortexA76CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA76CT.global_debug_rom	debug_rom
ARMCortexA76CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-293: ARMCortexA76CT MTI instances**

InstanceName	ComponentName
ARMCortexA76CT.AMU	PVBusLogger
ARMCortexA76CT.AMU.mapper	PVBusMapper
ARMCortexA76CT.DAP	PVBusLogger
ARMCortexA76CT.DAP.mapper	PVBusMapper
ARMCortexA76CT.DSU	DSU
ARMCortexA76CT.DSU.l3_flusher	AsyncCacheFlushUnit



InstanceName	ComponentName
ARMCortexA76CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA76CT.DSU.shared_cache	PVCache
ARMCortexA76CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA76CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA76CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA76CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA76CT.MMAP	PVBusLogger
ARMCortexA76CT.MMAP.mapper	PVBusMapper
ARMCortexA76CT.RAS	PVBusLogger
ARMCortexA76CT.RAS.mapper	PVBusMapper
ARMCortexA76CT.cpu0	ARM_Cortex-A76
ARMCortexA76CT.cpu0.UTLB	TLB
ARMCortexA76CT.cpu0.l1dcache	PVCache
ARMCortexA76CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA76CT.cpu0.l1icache	PVCache
ARMCortexA76CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA76CT.cpu0.l2cache	PVCache
ARMCortexA76CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA76CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA76CT.ext_bus	PVBusLogger
ARMCortexA76CT.ext_bus.mapper	PVBusMapper
ARMCortexA76CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA76CT

**Table 3-294: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[4]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable[4]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	These signals relate to core power down.
dbgprupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.

Name	Protocol	Type	Description
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core virtual System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA76CT

### **BROADCASTATOMIC**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### **BROADCASTCACHEMAINT**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### **BROADCASTOUTER**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

### **BROADCASTPERSIST**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CFGTE`****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**`cpuX.CRYPTODISABLE`****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-size****Type**

int



**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`dcache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`default_opmode`****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.



**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x100000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

**`periph_address_start`****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

**`ptw_latency`****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**`tlb_latency`****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**`tlbi_stall_enabled`****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat-dcache-cmos-to-pou-as-nop**

**Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.29 ARMCortexA77CT**

ARMCortexA77CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-295: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA77CT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.



The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGWRDUP and DBGRSTREQ are not implemented, but DBGWRUPREQ and DBGNOFWRDWN are implemented.
- Cache stashing capability.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM CortexA77CT

This model has the following Iris instances:

**Table 3-296: ARM CortexA77CT Iris instances**

InstanceName	ComponentName
ARM CortexA77CT	Cluster_ARM_Cortex-A77
ARM CortexA77CT.AMU	PVBusLogger
ARM CortexA77CT.AMU.mapper	PVBusMapper
ARM CortexA77CT.DAP	PVBusLogger
ARM CortexA77CT.DAP.mapper	PVBusMapper
ARM CortexA77CT.DSU	DSU
ARM CortexA77CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM CortexA77CT.DSU.mpam_busslave	PVBusSlave
ARM CortexA77CT.DSU.shared_cache	PVCache
ARM CortexA77CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM CortexA77CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM CortexA77CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM CortexA77CT.DSU.shared_cache.upstream[3]	PVBusSlave

InstanceName	ComponentName
ARMCortexA77CT.MMAP	PVBusLogger
ARMCortexA77CT.MMAP.mapper	PVBusMapper
ARMCortexA77CT.RAS	PVBusLogger
ARMCortexA77CT.RAS.mapper	PVBusMapper
ARMCortexA77CT.cpu0	ARM_Cortex-A77
ARMCortexA77CT.cpu0.UTLB	TLB
ARMCortexA77CT.cpu0.debug_rom	debug_rom
ARMCortexA77CT.cpu0.dtlb	TLB
ARMCortexA77CT.cpu0.l1dcache	PVCache
ARMCortexA77CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA77CT.cpu0.l1icache	PVCache
ARMCortexA77CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA77CT.cpu0.l2cache	PVCache
ARMCortexA77CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA77CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA77CT.ext_bus	PVBusLogger
ARMCortexA77CT.ext_bus.mapper	PVBusMapper
ARMCortexA77CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA77CT.global_debug_rom	debug_rom
ARMCortexA77CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-297: ARMCortexA77CT MTI instances**

InstanceName	ComponentName
ARMCortexA77CT.AMU	PVBusLogger
ARMCortexA77CT.AMU.mapper	PVBusMapper
ARMCortexA77CT.DAP	PVBusLogger
ARMCortexA77CT.DAP.mapper	PVBusMapper
ARMCortexA77CT.DSU	DSU
ARMCortexA77CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA77CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA77CT.DSU.shared_cache	PVCache
ARMCortexA77CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA77CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA77CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA77CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA77CT.MMAP	PVBusLogger
ARMCortexA77CT.MMAP.mapper	PVBusMapper
ARMCortexA77CT.RAS	PVBusLogger

InstanceName	ComponentName
ARMCortexA77CT.RAS.mapper	PVBusMapper
ARMCortexA77CT.cpu0	ARM_Cortex-A77
ARMCortexA77CT.cpu0.UTLB	TLB
ARMCortexA77CT.cpu0.l1dcache	PVCache
ARMCortexA77CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA77CT.cpu0.l1icache	PVCache
ARMCortexA77CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA77CT.cpu0.l2cache	PVCache
ARMCortexA77CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA77CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA77CT.ext_bus	PVBusLogger
ARMCortexA77CT.ext_bus.mapper	PVBusMapper
ARMCortexA77CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA77CT

**Table 3-298: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[4]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[4]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.

Name	Protocol	Type	Description
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA77CT

### BROADCASTATOMIC

#### Type

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CFGTE`****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**`cpuX.CRYPTODISABLE`****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**`cpuX.RVBARADDR`****Type**

int



**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**`cpuX.semihosting-T32_HLT`****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**`cpuX.semihosting-Thumb_SVC`****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**`cpuX.semihosting-cmd_line`****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**`cpuX.semihosting-cwd`****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**`cpuX.semihosting-enable`****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.



**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit

or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

### **l3cache-read\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

### **l3cache-size**

#### **Type**

int

#### **Default value**

0x100000

#### **Description**

L3 Cache size in bytes.

### **l3cache-snoop\_data\_transfer\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

### **l3cache-snoop\_issue\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).



**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.30 **ARMCortexA78AECT**

ARMCortexA78AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-299: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA78AECT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers’ view simulation, for example:
  - Automatic CPU retention mode.

- Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- Split/Lock is supported but with the limitations described in the AE-specific features implemented section.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpu`., where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.
- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.

As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

### Iris and MTI instances for ARMCortexA78AECT

This model has the following Iris instances:

**Table 3-300: ARMCortexA78AECT Iris instances**

InstanceName	ComponentName
ARMCortexA78AECT	Cluster_ARM_Cortex-A78AE
ARMCortexA78AECT.AMU	PVBusLogger
ARMCortexA78AECT.AMU.mapper	PVBusMapper
ARMCortexA78AECT.DAP	PVBusLogger

InstanceName	ComponentName
ARMCortexA78AECT.DAP.mapper	PVBusMapper
ARMCortexA78AECT.DSU	DSU
ARMCortexA78AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache	PVCache
ARMCortexA78AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA78AECT.MMAP	PVBusLogger
ARMCortexA78AECT.MMAP.mapper	PVBusMapper
ARMCortexA78AECT.RAS	PVBusLogger
ARMCortexA78AECT.RAS.mapper	PVBusMapper
ARMCortexA78AECT.cpu0	ARM_Cortex-A78AE
ARMCortexA78AECT.cpu0.UTLB	TLB
ARMCortexA78AECT.cpu0.debug_rom	debug_rom
ARMCortexA78AECT.cpu0.dtlb	TLB
ARMCortexA78AECT.cpu0.l1dcache	PVCache
ARMCortexA78AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu0.l1licache	PVCache
ARMCortexA78AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu0.l2cache	PVCache
ARMCortexA78AECT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA78AECT.cpu1	ARM_Cortex-A78AE
ARMCortexA78AECT.cpu1.UTLB	TLB
ARMCortexA78AECT.cpu1.debug_rom	debug_rom
ARMCortexA78AECT.cpu1.dtlb	TLB
ARMCortexA78AECT.cpu1.l1dcache	PVCache
ARMCortexA78AECT.cpu1.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu1.l1licache	PVCache
ARMCortexA78AECT.cpu1.l1licache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu1.l2cache	PVCache
ARMCortexA78AECT.cpu1.l2cache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu1.l2cache.upstream[1]	PVBusSlave
ARMCortexA78AECT.ext_bus	PVBusLogger
ARMCortexA78AECT.ext_bus.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA78AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA78AECT.global_debug_rom	debug_rom
ARMCortexA78AECT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-301: ARMCortexA78AECT MTI instances**

InstanceName	ComponentName
ARMCortexA78AECT.AMU	PVBusLogger
ARMCortexA78AECT.AMU.mapper	PVBusMapper
ARMCortexA78AECT.DAP	PVBusLogger
ARMCortexA78AECT.DAP.mapper	PVBusMapper
ARMCortexA78AECT.DSU	DSU
ARMCortexA78AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache	PVCache
ARMCortexA78AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA78AECT.MMAP	PVBusLogger
ARMCortexA78AECT.MMAP.mapper	PVBusMapper
ARMCortexA78AECT.RAS	PVBusLogger
ARMCortexA78AECT.RAS.mapper	PVBusMapper
ARMCortexA78AECT.cpu0	ARM_Cortex-A78AE
ARMCortexA78AECT.cpu0.UTLB	TLB
ARMCortexA78AECT.cpu0.l1dcache	PVCache
ARMCortexA78AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu0.l1licache	PVCache
ARMCortexA78AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu0.l2cache	PVCache
ARMCortexA78AECT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA78AECT.cpu1	ARM_Cortex-A78AE
ARMCortexA78AECT.cpu1.UTLB	TLB
ARMCortexA78AECT.cpu1.l1dcache	PVCache
ARMCortexA78AECT.cpu1.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu1.l1licache	PVCache

InstanceName	ComponentName
ARMCortexA78AECT.cpu1.l1icache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu1.l2cache	PVCache
ARMCortexA78AECT.cpu1.l2cache.upstream[0]	PVBusSlave
ARMCortexA78AECT.cpu1.l2cache.upstream[1]	PVBusSlave
ARMCortexA78AECT.ext_bus	PVBusLogger
ARMCortexA78AECT.ext_bus.mapper	PVBusMapper
ARMCortexA78AECT.gic_cpuif_decoder_cluster	GlCv3CPUInterfaceDecoder

## Ports for ARMCortexA78AECT

**Table 3-302: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamiQ pmu irq
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[4]	Signal	Master	Timer signals to SOC.

Name	Protocol	Type	Description
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[4]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[4]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.

Name	Protocol	Type	Description
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA78AECT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.



**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are

broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x2

**Description**

Number of cores per cluster.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-

read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

### **cpuX.l2cache-read\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

### **cpuX.l2cache-size**

#### **Type**

int

#### **Default value**

0x80000

#### **Description**

L2 Cache size in bytes.

### **cpuX.l2cache-snoop\_data\_transfer\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

### **cpuX.l2cache-snoop\_issue\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int



**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**`cpuX.semihosting-stack_limit`****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**`cpuX.trace_special_hlt_imm16`****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

**`cpuX.vfp-enable_at_reset`****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**`dcache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_lock\_step****Type**

bool

**Default value**

0x0

**Description**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_coherent\_icache****Type**

bool

**Default value**

0x0

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.



**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x100000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.31 ARMCortexA78CCT

ARMCortexA78CCT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-303: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About ARMCortexA78CCT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGPRUPREQ and DBGPRSTREQ are not implemented, but DBGPRUPREQ and DBGPRPWRDWN are implemented.
- Cache stashing capability.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where n identifies the core (0-7).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMCortexA78CCT

This model has the following Iris instances:

**Table 3-304: ARMCortexA78CCT Iris instances**

InstanceName	ComponentName
ARMCortexA78CCT	Cluster_ARM_Cortex-A78C
ARMCortexA78CCT.AMU	PVBusLogger
ARMCortexA78CCT.AMU.mapper	PVBusMapper
ARMCortexA78CCT.DAP	PVBusLogger
ARMCortexA78CCT.DAP.mapper	PVBusMapper
ARMCortexA78CCT.DSU	DSU
ARMCortexA78CCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78CCT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78CCT.DSU.shared_cache	PVCache
ARMCortexA78CCT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA78CCT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA78CCT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA78CCT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA78CCT.MMAP	PVBusLogger
ARMCortexA78CCT.MMAP.mapper	PVBusMapper
ARMCortexA78CCT.RAS	PVBusLogger
ARMCortexA78CCT.RAS.mapper	PVBusMapper
ARMCortexA78CCT.cpu0	ARM_Cortex-A78C
ARMCortexA78CCT.cpu0.UTLB	TLB
ARMCortexA78CCT.cpu0.debug_rom	debug_rom
ARMCortexA78CCT.cpu0.dtlb	TLB
ARMCortexA78CCT.cpu0.l1dcache	PVCache
ARMCortexA78CCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78CCT.cpu0.l1icache	PVCache
ARMCortexA78CCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA78CCT.cpu0.l2cache	PVCache
ARMCortexA78CCT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA78CCT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA78CCT.ext_bus	PVBusLogger
ARMCortexA78CCT.ext_bus.mapper	PVBusMapper
ARMCortexA78CCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA78CCT.global_debug_rom	debug_rom

InstanceName	ComponentName
ARMCortexA78CCT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-305: ARMCortexA78CCT MTI instances**

InstanceName	ComponentName
ARMCortexA78CCT.AMU	PVBusLogger
ARMCortexA78CCT.AMU.mapper	PVBusMapper
ARMCortexA78CCT.DAP	PVBusLogger
ARMCortexA78CCT.DAP.mapper	PVBusMapper
ARMCortexA78CCT.DSU	DSU
ARMCortexA78CCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78CCT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78CCT.DSU.shared_cache	PVCache
ARMCortexA78CCT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA78CCT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA78CCT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA78CCT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA78CCT.MMAP	PVBusLogger
ARMCortexA78CCT.MMAP.mapper	PVBusMapper
ARMCortexA78CCT.RAS	PVBusLogger
ARMCortexA78CCT.RAS.mapper	PVBusMapper
ARMCortexA78CCT.cpu0	ARM_Cortex-A78C
ARMCortexA78CCT.cpu0.UTLB	TLB
ARMCortexA78CCT.cpu0.l1dcache	PVCache
ARMCortexA78CCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78CCT.cpu0.l1icache	PVCache
ARMCortexA78CCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA78CCT.cpu0.l2cache	PVCache
ARMCortexA78CCT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA78CCT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA78CCT.ext_bus	PVBusLogger
ARMCortexA78CCT.ext_bus.mapper	PVBusMapper
ARMCortexA78CCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA78CCT

**Table 3-306: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.



Name	Protocol	Type	Description
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[8]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[8]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[8]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[8]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[8]	Signal	Master	Timer signals to SOC.
commirq[8]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[8]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[8]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[8]	Signal	Slave	Disable cryptography extensions after reset.
cti[8]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.

Name	Protocol	Type	Description
ctidbgirq[8]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[8]	Signal	Master	No power-down request.
dbgpwrupreq[8]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[8]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[8]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[8]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[8]	PChannel	Slave	PChannels for cores
pmbirq[8]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[8]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[8]	Signal	Slave	Per core RAM Error Interrupt.
reset[8]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[8]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[8]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[8]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.

Name	Protocol	Type	Description
vcpumntirq[8]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[8]	Signal	Slave	Virtualised FIQ.
vinithi[8]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[8]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[8]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A78CCT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTOUTER

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`cpuX.l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.



**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**`cpuX.semihosting-ARM_SVC`****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**`cpuX.semihosting-T32_HLT`****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**`cpuX.semihosting-Thumb_SVC`****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**`cpuX.semihosting-cmd_line`****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**`cpuX.semihosting-cwd`****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool



**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_coherent\_icache****Type**

bool

**Default value**

0x0

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l3cache-size`****Type**

int

**Default value**

0x100000

**Description**

L3 Cache size in bytes.

**`l3cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l3cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l3cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

### **l3cache-write\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

### **pchannel\_treat\_simreset\_as\_poreset**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Register core as ON state to cluster with simulation reset.

### **periph\_address\_end**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

### **periph\_address\_start**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.32 **ARMCortexA78CT**

ARMCortexA78CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-307: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA78CT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers’ view simulation, for example:



- Automatic CPU retention mode.
- Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARMCortexA78CT

This model has the following Iris instances:

**Table 3-308: ARMCortexA78CT Iris instances**

InstanceName	ComponentName
ARMCortexA78CT	Cluster_ARM_Cortex-A78
ARMCortexA78CT.AMU	PVBusLogger
ARMCortexA78CT.AMU.mapper	PVBusMapper
ARMCortexA78CT.DAP	PVBusLogger
ARMCortexA78CT.DAP.mapper	PVBusMapper
ARMCortexA78CT.DSU	DSU
ARMCortexA78CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78CT.DSU.shared_cache	PVCache
ARMCortexA78CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA78CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA78CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA78CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA78CT.MMAP	PVBusLogger
ARMCortexA78CT.MMAP.mapper	PVBusMapper
ARMCortexA78CT.RAS	PVBusLogger
ARMCortexA78CT.RAS.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA78CT.cpu0	ARM_Cortex-A78
ARMCortexA78CT.cpu0.UTLB	TLB
ARMCortexA78CT.cpu0.debug_rom	debug_rom
ARMCortexA78CT.cpu0.dtlb	TLB
ARMCortexA78CT.cpu0.l1dcache	PVCache
ARMCortexA78CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78CT.cpu0.l1icache	PVCache
ARMCortexA78CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA78CT.cpu0.l2cache	PVCache
ARMCortexA78CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA78CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA78CT.ext_bus	PVBusLogger
ARMCortexA78CT.ext_bus.mapper	PVBusMapper
ARMCortexA78CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA78CT.global_debug_rom	debug_rom
ARMCortexA78CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-309: ARMCortexA78CT MTI instances**

InstanceName	ComponentName
ARMCortexA78CT.AMU	PVBusLogger
ARMCortexA78CT.AMU.mapper	PVBusMapper
ARMCortexA78CT.DAP	PVBusLogger
ARMCortexA78CT.DAP.mapper	PVBusMapper
ARMCortexA78CT.DSU	DSU
ARMCortexA78CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78CT.DSU.shared_cache	PVCache
ARMCortexA78CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA78CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA78CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA78CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA78CT.MMAP	PVBusLogger
ARMCortexA78CT.MMAP.mapper	PVBusMapper
ARMCortexA78CT.RAS	PVBusLogger
ARMCortexA78CT.RAS.mapper	PVBusMapper
ARMCortexA78CT.cpu0	ARM_Cortex-A78
ARMCortexA78CT.cpu0.UTLB	TLB
ARMCortexA78CT.cpu0.l1dcache	PVCache

InstanceName	ComponentName
ARMCortexA78CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78CT.cpu0.l1icache	PVCache
ARMCortexA78CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA78CT.cpu0.l2cache	PVCache
ARMCortexA78CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA78CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA78CT.ext_bus	PVBusLogger
ARMCortexA78CT.ext_bus.mapper	PVBusMapper
ARMCortexA78CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA78CT

**Table 3-310: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[4]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq

Name	Protocol	Type	Description
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[4]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[4]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.

Name	Protocol	Type	Description
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A78CT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.



**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`dcache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit

or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

### **`dcache-read_latency`**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

### **`dcache-size`**

#### **Type**

int

#### **Default value**

0x10000

#### **Description**

L1 D-Cache size in bytes.

### **`dcache-snoop_data_transfer_latency`**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

### **`dcache-state_modelled`**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Set whether D-cache has stateful implementation.



**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_coherent\_icache****Type**

bool

**Default value**

0x0

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**`l3cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`l3cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`l3cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x100000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.



**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**`treat-dcache-cmos-to-pou-as-nop`**

**Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**`walk_cache_latency`**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.33 ARMCortexA320CT**

ARMCortexA320CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-311: IP revisions support**

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA320CT**

The number of cores currently supported is 1,2,4.

**Limitations**

The following features are not yet supported, and will be added in a future release:

- No support for transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (TLM2 extensions from Arm) bus protocols yet.
- No support for Core-Complex yet.

- BROADCASTCACHEMAINTPOU pin is not implemented.
- COREINSTRET and COREINSTRRUN signals are not implemented.
- MEM\_RET power mode is not supported.

The following features are not supported in this or future releases:

- DynamIQ features that are negligible to the programmers view simulation.
- 256-bit wide output transactions.
- Error correction/detection features.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.

### Iris and MTI instances for ARM CortexA320CT

This model has the following Iris instances:

**Table 3-312: ARM CortexA320CT Iris instances**

InstanceName	ComponentName
ARM CortexA320CT	Cluster_ARM_Cortex-A320
ARM CortexA320CT.AMU	PVBusLogger
ARM CortexA320CT.AMU.mapper	PVBusMapper
ARM CortexA320CT.DAP	PVBusLogger
ARM CortexA320CT.DAP.mapper	PVBusMapper
ARM CortexA320CT.DSU	DSU
ARM CortexA320CT.DSU.PPU_cluster	PPUv1
ARM CortexA320CT.DSU.PPU_cluster.busslave	PVBusSlave
ARM CortexA320CT.DSU.PPU_core0	PPUv1
ARM CortexA320CT.DSU.PPU_core0.busslave	PVBusSlave
ARM CortexA320CT.DSU.mpam_busslave	PVBusSlave
ARM CortexA320CT.DSU.shared_cache	PVCache
ARM CortexA320CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM CortexA320CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM CortexA320CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM CortexA320CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARM CortexA320CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARM CortexA320CT.DSU.utility_slave[0]	PVBusSlave
ARM CortexA320CT.MMAP	PVBusLogger
ARM CortexA320CT.MMAP.mapper	PVBusMapper
ARM CortexA320CT.RAS	PVBusLogger

InstanceName	ComponentName
ARMCortexA320CT.RAS.mapper	PVBusMapper
ARMCortexA320CT.cpu0	ARM_Cortex-A320
ARMCortexA320CT.cpu0.UTLB	TLB
ARMCortexA320CT.cpu0.debug_rom	debug_rom
ARMCortexA320CT.cpu0.dtlb	TLB
ARMCortexA320CT.cpu0.l1dcache	PVCache
ARMCortexA320CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA320CT.cpu0.l1icache	PVCache
ARMCortexA320CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA320CT.ext_bus	PVBusLogger
ARMCortexA320CT.ext_bus.mapper	PVBusMapper
ARMCortexA320CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA320CT.global_debug_rom	debug_rom
ARMCortexA320CT.l2_cache	PVCache
ARMCortexA320CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexA320CT.secondary_debug_rom	debug_rom
ARMCortexA320CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-313: ARMCortexA320CT MTI instances**

InstanceName	ComponentName
ARMCortexA320CT.AMU	PVBusLogger
ARMCortexA320CT.AMU.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA320CT.DAP	PVBusLogger
ARMCortexA320CT.DAP.mapper	PVBusMapper
ARMCortexA320CT.DSU	DSU
ARMCortexA320CT.DSU.PPU_cluster	PPUv1
ARMCortexA320CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA320CT.DSU.PPU_core0	PPUv1
ARMCortexA320CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA320CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA320CT.DSU.shared_cache	PVCache
ARMCortexA320CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA320CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA320CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA320CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA320CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA320CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA320CT.MMAP	PVBusLogger
ARMCortexA320CT.MMAP.mapper	PVBusMapper
ARMCortexA320CT.RAS	PVBusLogger
ARMCortexA320CT.RAS.mapper	PVBusMapper
ARMCortexA320CT.cpu0	ARM_Cortex-A320
ARMCortexA320CT.cpu0.UTLB	TLB
ARMCortexA320CT.cpu0.l1dcache	PVCache
ARMCortexA320CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA320CT.cpu0.l1icache	PVCache
ARMCortexA320CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA320CT.ext_bus	PVBusLogger
ARMCortexA320CT.ext_bus.mapper	PVBusMapper
ARMCortexA320CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA320CT.l2_cache	PVCache
ARMCortexA320CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[2]	PVBusSlave

InstanceName	ComponentName
ARMCortexA320CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexA320CT.l2_cache.upstream[9]	PVBusSlave

## Ports for ARMCortexA320CT

**Table 3-314: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP subordinate port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsmpchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[4]	Signal	Master	Timer signals to SOC
CNTHVIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[4]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
complexerrirq[1]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
complexfaultirq[1]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
core_clk_in[4]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel[4]	PChannel	Master	Core PCSM signals
core_powerdown_out[4]	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
coreerrirq[4]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[4]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.

Name	Protocol	Type	Description
dbgpwrupreq[4]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main manager interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[4]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[4]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[4]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.



Name	Protocol	Type	Description
trbirq[4]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus subordinate
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA320CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAIN****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x80000

**Description**

Value of CPU Configuration Register.

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x0

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_cache\_protection****Type**

int

**Default value**

0x1

**Description**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int



**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**`dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`dcache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`enable_simulation_performance_optimizations`****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.



**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**has\_actlr2****Type**

bool

**Default value**

0x1

**Description**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**`l2cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**ras\_extra\_configurations****Type**

string

**Default value**

```
"[ { \"Index\": 0, \"ERXMISCO_mask\": 0xFFFFC0003FCF, \"ERXMISC1_mask\": 0x03F87000FFF30F07, \"ERXPFGCTL_reset\": 0x1000}, { \"Index\": 1, \"ERXMISCO_mask\": 0xFFFFFE007FFCF, \"ERXMISCO_reset\": 0x2, \"ERXSTATUS_IERR_mask\": 0x300, \"ERXMISC1_mask\": 0x0FF8700FFFF31F0F, \"ERXPFGCTL_reset\": 0x1000} ]"
```

**Description**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCn masks - these are 64 bit masks covering the 64 bit registers ERXMISCn\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

**ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**revision\_number****Type**

int

**Default value**

0x0

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.



**store\_excl\_fail\_tag\_check\_action****Type**

int

**Default value**

0x0

**Description**

Behavior of tag check by core when a store exclusive fails. 0, Tag check ignored, 1, Tag check done if exclusive fails by global monitor.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.34 ARMCortexA510CT

ARMCortexA510CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-315: IP revisions support**

Revision	Quality level
r0p0	Full support
r1p3	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA510CT**

The model supports the following features:

- DynamIQ Shared Unit-110 (DSU-110) system registers.
- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- PChannel for the cluster and for each core.
- BROADCASTPERSIST pin.
- Optional peripheral port.
- L3Cache partition.
- Per-core clock.
- Utility bus.
- Revision R1 is the default configuration, with 32-bit support at EL0. R1 supports both configurations of EL0, with or without A32 support. For 64-bit only mode, set parameter `max_32bit_el=-1`.
- To configure revision R0, set parameter `revision_number=0`.

Support for the following features is planned for a future release:

- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (the Arm TLM2 extensions) bus protocols.
- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, and nPMBIRQ signals.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- Error correction or detection features.
- Self-test features (MBIST).
- Snoop filtering.
- Latency configuration.
- Cache stashing capability.

This model supports the Arm®v8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARMCortexA510CT

This model has the following Iris instances:

**Table 3-316: ARMCortexA510CT Iris instances**

InstanceName	ComponentName
ARMCortexA510CT	Cluster_ARM_Cortex-A510
ARMCortexA510CT.AMU	PVBusLogger
ARMCortexA510CT.AMU.mapper	PVBusMapper
ARMCortexA510CT.DAP	PVBusLogger
ARMCortexA510CT.DAP.mapper	PVBusMapper
ARMCortexA510CT.DSU	DSU
ARMCortexA510CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA510CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA510CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT.DSU.shared_cache	PVCache
ARMCortexA510CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA510CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA510CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA510CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA510CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT.MMAP	PVBusLogger
ARMCortexA510CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT.RAS	PVBusLogger

InstanceName	ComponentName
ARMCortexA510CT.RAS.mapper	PVBusMapper
ARMCortexA510CT.cpu0	ARM_Cortex-A510
ARMCortexA510CT.cpu0.UTLB	TLB
ARMCortexA510CT.cpu0.debug_rom	debug_rom
ARMCortexA510CT.cpu0.dtlb	TLB
ARMCortexA510CT.cpu0.l1dcache	PVCache
ARMCortexA510CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT.cpu0.l1icache	PVCache
ARMCortexA510CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA510CT.cpu0.l2cache	PVCache
ARMCortexA510CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT.ext_bus	PVBusLogger
ARMCortexA510CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT.global_debug_rom	debug_rom
ARMCortexA510CT.secondary_debug_rom	debug_rom
ARMCortexA510CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-317: ARMCortexA510CT MTI instances**

InstanceName	ComponentName
ARMCortexA510CT.AMU	PVBusLogger
ARMCortexA510CT.AMU.mapper	PVBusMapper
ARMCortexA510CT.DAP	PVBusLogger
ARMCortexA510CT.DAP.mapper	PVBusMapper
ARMCortexA510CT.DSU	DSU
ARMCortexA510CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA510CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA510CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT.DSU.shared_cache	PVCache
ARMCortexA510CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA510CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA510CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA510CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA510CT.DSU.utility_slave[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexA510CT.MMAP	PVBusLogger
ARMCortexA510CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT.RAS	PVBusLogger
ARMCortexA510CT.RAS.mapper	PVBusMapper
ARMCortexA510CT.cpu0	ARM_Cortex-A510
ARMCortexA510CT.cpu0.UTLB	TLB
ARMCortexA510CT.cpu0.l1dcache	PVCache
ARMCortexA510CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT.cpu0.l1licache	PVCache
ARMCortexA510CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT.cpu0.l2cache	PVCache
ARMCortexA510CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT.ext_bus	PVBusLogger
ARMCortexA510CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA510CT

**Table 3-318: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[12]	Signal	Slave	This signal is for EE bit initialisation.

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[12]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[12]	Signal	Master	Timer signals to SOC
CNTHVIRQ[12]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[12]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[12]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[12]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[12]	Signal	Master	Timer signals to SOC.
commirq[12]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[12]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel[12]	PChannel	Master	Core PCSM signals
core_powerdown_out[12]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[12]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[12]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[12]	Signal	Slave	Disable cryptography extensions after reset.

Name	Protocol	Type	Description
cti[12]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[12]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[12]	Signal	Master	No power-down request.
dbgpwrupreq[12]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[12]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[12]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[12]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmuirq[12]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[12]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[12]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[12]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[12]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[12]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.

Name	Protocol	Type	Description
ticks[12]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[12]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[12]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[12]	Signal	Slave	Virtualised FIQ.
virq[12]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[12]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA510CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0



**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`**

**Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`**

**Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CRYPTODISABLE`**

**Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**`cpuX.RVBARADDR`**

**Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**`cpuX.enable_trace_special_hlt_imm16`**

**Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter `trace_special_hlt_imm16`.

**`cpuX.l2cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x20

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0



**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**`dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.



**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.



**max\_32bit\_el****Type**

int

**Default value**

0x0

**Description**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: ELO supports A32. This parameter is ignored in Rev0 i.e. when revision\_number = 0.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**patch\_level****Type**

int

**Default value**

0x3

**Description**

Patch level of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Revision field in MIDR/MIDR\_EL1. Corresponds to the patch number Y in rXpY.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**revision\_number****Type**

int

**Default value**

0x1

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.35 ARMCortexA510CT\_CortexA710CT

ARMCortexA510CT\_CortexA710CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-319: IP revisions support**

Revision	Quality level
CortexA510 r0p0	Full support
CortexA510 r1p3	Full support
CortexA710 r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About ARMCortexA510CT\_CortexA710CT

The number of cores in each subcluster is configurable using the following parameters:

#### **subcluster0.NUM\_CORES**

Possible values are 1-11 (ARMCortexA510CT).

#### **subcluster1.NUM\_CORES**

Possible values are 1-11 (ARMCortexA710CT).

The total number of cores in the cluster cannot exceed 12.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-10] for cores in subcluster0.
- <port\_name>[11-21] for cores in subcluster1.



#### Note

All instances in the Master cross trigger matrix port array `cti[22]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu10` identify cores in subcluster0.
- `subcluster1.cpu0` to `subcluster1.cpu10` identify cores in subcluster1.

For information about the cores in this model, see:

- [ARMCortexA510CT](#).
- [ARMCortexA710CT](#).

### Iris and MTI instances for ARMCortexA510CT\_CortexA710CT

This model has the following Iris instances:

**Table 3-320: ARMCortexA510CT\_CortexA710CT Iris instances**

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT	Cluster_ ARM_Cortex510_CortexA710_Heterogeneous

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA710CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA710CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.DSU	DSU
ARMCortexA510CT_CortexA710CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT_CortexA710CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.PPU_core1	PPUv1
ARMCortexA510CT_CortexA710CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA710CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA710CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.cpu0.debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT.cpu1.debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT.global_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT.secondary_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT.subcluster0	Subcluster_ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.dtlb	TLB
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l1icache	PVCache

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster0.sve	ScalableVectorExtension
ARMCortexA510CT_CortexA710CT.subcluster1	Subcluster_ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0	ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster1.cpu1.dtlb	TLB
ARMCortexA510CT_CortexA710CT.subcluster1.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-321: ARMCortexA510CT\_CortexA710CT MTI instances**

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA710CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA710CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.DSU	DSU
ARMCortexA510CT_CortexA710CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT_CortexA710CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.PPU_core1	PPUv1
ARMCortexA510CT_CortexA710CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache	PVCache

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA710CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA710CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0	ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave

## Ports for ARMCortexA510CT\_CortexA710CT

**Table 3-322: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDOMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).

Name	Protocol	Type	Description
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[22]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[22]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[22]	Signal	Master	Timer signals to SOC
CNTHVIRQ[22]	Signal	Master	Timer signals to SOC.



Name	Protocol	Type	Description
CNTHVSIRQ[22]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[22]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[22]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[22]	Signal	Master	Timer signals to SOC.
commirq[22]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[22]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[22]	PChannel	Master	Core PCSM signals
core_powerdown_out[22]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[22]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[22]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[22]	Signal	Slave	Disable cryptography extensions after reset.
cti[22]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[22]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[22]	Signal	Master	No power-down request.
dbgpwrupreq[22]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[22]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[22]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[22]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmuirq[22]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.

Name	Protocol	Type	Description
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[22]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[22]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[22]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[22]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[22]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[22]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[22]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[22]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[22]	Signal	Slave	Virtualised FIQ.
virq[22]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[22]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA510CT\_CortexA710CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

**Description**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMPO input signal).

**AEND1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

**AEND2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMPO input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x0

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0



**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**l3cache-size****Type**

int

**Default value**

0x0

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l3cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l3cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`l3cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`mpmm_accumulator_multiplier`****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of  $n$  means the accumulator will use  $(n * \text{accumulator value})$  to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**subcluster0.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied.

This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x20

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0



**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**`subcluster0.dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**subcluster0.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**subcluster0.etc.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster0.etc.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster0.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster0.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster0.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster0.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster0.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster0.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster0.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster0.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.



**subcluster0.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster0.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster0.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster0.has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`subcluster0.icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster0.icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`subcluster0.invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**`subcluster0.max_32bit_el`****Type**

int

**Default value**

0x0

**Description**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: ELO supports A32. This parameter is ignored in Rev0 i.e. when revision\_number = 0.

**subcluster0.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**subcluster0.revision\_number****Type**

int

**Default value**

0x1

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster1.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0



**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster1.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.max_code_cache_mb`****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.ete.CLAIMTAGS****Type**

int

**Default value**

0x20

**Description**

Number of claim tags.

**subcluster1.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster1.ete.NumberOfRSPairs****Type**

int



**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster1.etc.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster1.etc.PIDR\_REVAND****Type**

int

**Default value**

0x1

**Description**

TRCPIDR REVAND value.

**subcluster1.etc.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster1.etc.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster1.etc.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster1.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster1.ete.REVISION****Type**

int

**Default value**

0x2

**Description**

TRCIDR1 revision value.

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster1.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster1.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster1.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster1.ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**subcluster1.ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**subcluster1.ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**subcluster1.ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect.  
1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster1.has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster1.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster1.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.

Note: Enabling this parameter will reduce simulation performance.

**subcluster1.memory\_tagging\_support\_level****Type**

int

**Default value**

0x2

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

**subcluster1.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled**

**Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat-dcache-cmos-to-pou-as-nop**

**Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**subcluster1.walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.36 ARMCortexA510CT\_CortexA710CT\_CortexX2CT**

ARMCortexA510CT\_CortexA710CT\_CortexX2CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-323: IP revisions support**

Revision	Quality level
CortexA510 r0p0	Full support
CortexA510 r1p3	Full support
CortexA710 r2p0	Full support



Revision	Quality level
CortexX2 r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About ARMCortexA510CT\_CortexA710CT\_CortexX2CT

The number of cores in each subcluster is configurable using the following parameters:

### **subcluster0.NUM\_CORES**

Possible values are 1-10 (ARMCortexA510CT).

### **subcluster1.NUM\_CORES**

Possible values are 1-10 (ARMCortexA710CT).

### **subcluster2.NUM\_CORES**

Possible values are 1-10 (ARMCortexX2CT).

The total number of cores in the cluster cannot exceed 12.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- `<port_name>[0-9]` for cores in `subcluster0`.
- `<port_name>[10-19]` for cores in `subcluster1`.
- `<port_name>[20-29]` for cores in `subcluster2`.



### Note

All instances in the Master cross trigger matrix port array, `cti[30]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu9` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu9` identify cores in `subcluster1`.
- `subcluster2.cpu0` to `subcluster2.cpu9` identify cores in `subcluster2`.

For information about the cores in this model, see:

- [ARMCortexA510CT](#).
- [ARMCortexA710CT](#).
- [ARMCortexX2CT](#).

## Iris and MTI instances for ARMCortexA510CT\_CortexA710CT\_CortexX2CT

This model has the following Iris instances:

**Table 3-324: ARMCortexA510CT\_CortexA710CT\_CortexX2CT Iris instances**

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX2CT	Cluster_ARM_CortexA510_CortexA710_CortexX2_Heterogeneous
ARMCortexA510CT_CortexA710CT_CortexX2CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU	DSU
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core1	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core2	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core2.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[7]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.RAS	PVBusLogger

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX2CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.cpu0.debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX2CT.cpu1.debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX2CT.cpu2.debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX2CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT_CortexX2CT.global_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX2CT.secondary_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0	Subcluster_ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.dtlb	TLB
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.sve	ScalableVectorExtension
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1	Subcluster_ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0	ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu1.dtlb	TLB
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.sve	ScalableVectorExtension
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2	Subcluster_ARM_Cortex-X2
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0	ARM_Cortex-X2
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu2.dtlb	TLB
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-325: ARMCortexA510CT\_CortexA710CT\_CortexX2CT MTI instances**

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX2CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU	DSU
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core0.busslave	PVBusSlave

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core1	PPUV1
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core2	PPUV1
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_core2.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[7]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0	ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0	ARM_Cortex-X2
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l1icache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0.l2cache.upstream[1]	PVBusSlave

### Ports for ARMCortexA510CT\_CortexA710CT\_CortexX2CT

**Table 3-326: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[30]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.

Name	Protocol	Type	Description
cluster_pcs_m_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[30]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[30]	Signal	Master	Timer signals to SOC
CNTHVIRQ[30]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[30]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[30]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[30]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[30]	Signal	Master	Timer signals to SOC.
commirq[30]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[30]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel[30]	PChannel	Master	Core PCSM signals
core_powerdown_out[30]	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
coreerrirq[30]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[30]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[30]	Signal	Slave	Disable cryptography extensions after reset.
cti[30]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[30]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[30]	Signal	Master	No power-down request.

Name	Protocol	Type	Description
dbgpwrupreq[30]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[30]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[30]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[30]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmuirq[30]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[30]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[30]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[30]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[30]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[30]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[30]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[30]	Signal	Master	Interrupt signal from the trace buffer unit.



Name	Protocol	Type	Description
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[30]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[30]	Signal	Slave	Virtualised FIQ.
virq[30]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[30]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA510CT\_CortexA710CT\_CortexX2CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAIN****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x0

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x0

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int



**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**`periph_address_end`****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**`periph_address_start`****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**`subcluster0.CPUCFR`****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**`subcluster0.NUM_CORES`****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x20

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC**

**Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT**

**Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-Thumb\_SVC**

**Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line**

**Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd**

**Type**

string

**Default value**

""



**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**subcluster0.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster0.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster0.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster0.ete.PIDR\_CMOD****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CMOD value.

**subcluster0.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster0.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster0.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster0.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster0.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster0.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster0.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster0.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster0.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.



**subcluster0.has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster0.icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`subcluster0.invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**`subcluster0.max_32bit_el`****Type**

int

**Default value**

0x0

**Description**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: ELO supports A32. This parameter is ignored in Rev0 i.e. when `revision_number = 0`.

**`subcluster0.memory_tagging_support_level`****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**subcluster0.revision\_number****Type**

int

**Default value**

0x1

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter `trace_special_hlt_imm16`.

**`subcluster1.cpuX.l2cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-`

read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster1.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.



**subcluster1.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.etc.CLAIMTAGS****Type**

int

**Default value**

0x20

**Description**

Number of claim tags.

**subcluster1.etc.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster1.etc.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster1.etc.PIDR\_CMOD****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CMOD value.

**subcluster1.ete.PIDR\_REVAND****Type**

int

**Default value**

0x1

**Description**

TRCPIDR REVAND value.

**subcluster1.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster1.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster1.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster1.ete.RETSTACK****Type**

int

**Default value**

0x3



**Description**

Return stack depth.

**subcluster1.ete.REVISION****Type**

int

**Default value**

0x2

**Description**

TRCIDR1 revision value.

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster1.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster1.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster1.etc.TRCRSRTA\_FORCED\_EXCEP**

**Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster1.ext\_abort\_device\_read\_is\_sync**

**Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**subcluster1.ext\_abort\_device\_write\_is\_sync**

**Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**subcluster1.ext\_abort\_so\_read\_is\_sync**

**Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**subcluster1.ext\_abort\_so\_write\_is\_sync**

**Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster1.has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster1.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster1.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster1.memory\_tagging\_support\_level****Type**

int

**Default value**

0x2

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

**subcluster1.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**subcluster1.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

**subcluster2.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster2.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster2.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster2.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster2.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster2.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.



**subcluster2.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster2.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster2.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster2.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-A64\_HLT**

**Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-ARM\_SVC**

**Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster2.cpuX.semihosting-T32\_HLT**

**Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-Thumb\_SVC**

**Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster2.cpuX.semihosting-cmd\_line**

**Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster2.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster2.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster2.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster2.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster2.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster2.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster2.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster2.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster2.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster2.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.



**subcluster2.ete.CLAIMTAGS****Type**

int

**Default value**

0x20

**Description**

Number of claim tags.

**subcluster2.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster2.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster2.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster2.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster2.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster2.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster2.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster2.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster2.ete.REVISION****Type**

int

**Default value**

0x2

**Description**

TRCIDR1 revision value.

**subcluster2.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster2.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster2.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster2.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster2.ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.

**subcluster2.ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**subcluster2.ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**subcluster2.ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**subcluster2.ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**subcluster2.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster2.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster2.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster2.has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster2.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster2.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster2.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster2.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster2.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster2.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster2.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster2.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**subcluster2.memory\_tagging\_support\_level****Type**

int

**Default value**

0x2

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

**subcluster2.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster2.treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**subcluster2.walk\_cache\_latency****Type**

int

**Default value**

0x0



Description

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.37 ARMCortexA510CT\_CortexA710CT\_CortexX3CT

ARMCortexA510CT\_CortexA710CT\_CortexX3CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-327: IP revisions support

Revision	Quality level
CortexA510 r0p0	Full support
CortexA510 r1p3	Full support
CortexA710 r2p0	Full support
CortexX3 r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexA510CT\_CortexA710CT\_CortexX3CT

The number of cores in each subcluster is configurable using the following parameters:

- subcluster0.NUM\_CORES**  
Possible values are 1-10 (ARMCortexA510CT).
- subcluster1.NUM\_CORES**  
Possible values are 1-10 (ARMCortexA710CT).
- subcluster2.NUM\_CORES**  
Possible values are 1-10 (ARMCortexX3CT).

The total number of cores in the cluster cannot exceed 12.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-9] for cores in subcluster0.
- <port\_name>[10-19] for cores in subcluster1.
- <port\_name>[20-29] for cores in subcluster2.



All instances in the Master cross trigger matrix port array, `cti[30]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu9` identify cores in `subcluster0`.

- subcluster1.cpu0 to subcluster1.cpu9 identify cores in subcluster1.
- subcluster2.cpu0 to subcluster2.cpu9 identify cores in subcluster2.

For information about the cores in this model, see:

- [ARMCortexA510CT](#).
- [ARMCortexA710CT](#).
- [ARMCortexX3CT](#).

## Iris and MTI instances for ARMCortexA510CT\_CortexA710CT\_CortexX3CT

This model has the following Iris instances:

**Table 3-328: ARMCortexA510CT\_CortexA710CT\_CortexX3CT Iris instances**

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX3CT	Cluster_ARM_CortexA510_CortexA710_CortexX3_Heterogeneous
ARMCortexA510CT_CortexA710CT_CortexX3CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU	DSU
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core1	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core2	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core2.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[3]	PVBusSlave

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[7]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.cpu0.debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX3CT.cpu1.debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX3CT.cpu2.debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX3CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT_CortexX3CT.global_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX3CT.secondary_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0	Subcluster_ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.dtlb	TLB
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.sve	ScalableVectorExtension
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1	Subcluster_ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0	ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu1.dtlb	TLB
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.sve	ScalableVectorExtension
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2	Subcluster_ARM_Cortex-X3
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0	ARM_Cortex-X3
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu2.dtlb	TLB
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-329: ARMCortexA510CT\_CortexA710CT\_CortexX3CT MTI instances**

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX3CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU	DSU
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core1	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core2	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_core2.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[7]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l2cache	PVCache

InstanceName	ComponentName
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0	ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l1icache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0	ARM_Cortex-X3
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l1icache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0.l2cache.upstream[1]	PVBusSlave

## Ports for ARMCortexA510CT\_CortexA710CT\_CortexX3CT

**Table 3-330: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.

Name	Protocol	Type	Description
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[30]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[30]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[30]	Signal	Master	Timer signals to SOC
CNTHVIRQ[30]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[30]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[30]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[30]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[30]	Signal	Master	Timer signals to SOC.
commirq[30]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[30]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel[30]	PChannel	Master	Core PCSM signals
core_powerdown_out[30]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info

Name	Protocol	Type	Description
coreerrirq[30]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[30]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[30]	Signal	Slave	Disable cryptography extensions after reset.
cti[30]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[30]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[30]	Signal	Master	No power-down request.
dbgpwrupreq[30]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[30]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[30]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[30]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[30]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[30]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[30]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[30]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[30]	Signal	Slave	Per core RAM Error Interrupt.



Name	Protocol	Type	Description
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[30]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[30]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[30]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[30]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[30]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[30]	Signal	Slave	Virtualised FIQ.
virq[30]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[30]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM CortexA510CT\_CortexA710CT\_CortexX3CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

**AEND2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x0

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x0

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0



**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quanta in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**subcluster0.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x20

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.



**subcluster0.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**subcluster0.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**subcluster0.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit

or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

#### **`subcluster0.dcache-read_latency`**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

#### **`subcluster0.dcache-size`**

##### **Type**

int

##### **Default value**

0x10000

##### **Description**

L1 D-Cache size in bytes.

#### **`subcluster0.dcache-snoop_data_transfer_latency`**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

#### **`subcluster0.dcache-write_access_latency`**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster0.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster0.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster0.ete.PIDR\_CMOD****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CMOD value.

**subcluster0.ete.PIDR\_REVAND**

**Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster0.ete.PIDR\_REVISION**

**Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster0.ete.Q\_CADENCE**

**Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster0.ete.RES0\_STATEFUL**

**Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster0.ete.RETSTACK**

**Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster0.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster0.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster0.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster0.etc.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0



**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect.  
1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster0.has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster0.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster0.icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**subcluster0.max\_32bit\_el****Type**

int

**Default value**

0x0

**Description**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: ELO supports A32. This parameter is ignored in Rev0 i.e. when revision\_number = 0.

**subcluster0.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**subcluster0.revision\_number****Type**

int

**Default value**

0x1

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter `trace_special_hlt_imm16`.

**`subcluster1.cpuX.l2cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-`

read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

#### **subcluster1.cpuX.l2cache-read\_latency**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

#### **subcluster1.cpuX.l2cache-size**

##### **Type**

int

##### **Default value**

0x80000

##### **Description**

L2 Cache size in bytes.

#### **subcluster1.cpuX.l2cache-snoop\_data\_transfer\_latency**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

#### **subcluster1.cpuX.l2cache-snoop\_issue\_latency**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.



**subcluster1.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.ete.CLAIMTAGS****Type**

int

**Default value**

0x20

**Description**

Number of claim tags.

**subcluster1.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster1.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster1.ete.PIDR\_CMOD****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CMOD value.

**subcluster1.ete.PIDR\_REVAND****Type**

int

**Default value**

0x1

**Description**

TRCPIDR REVAND value.

**subcluster1.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster1.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster1.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster1.ete.RETSTACK****Type**

int

**Default value**

0x3



**Description**

Return stack depth.

**subcluster1.etc.REVISION****Type**

int

**Default value**

0x2

**Description**

TRCIDR1 revision value.

**subcluster1.etc.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster1.etc.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster1.etc.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster1.etc.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster1.etc.TRCRSRTA\_FORCED\_EXCEP**

**Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster1.ext\_abort\_device\_read\_is\_sync**

**Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**subcluster1.ext\_abort\_device\_write\_is\_sync**

**Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**subcluster1.ext\_abort\_so\_read\_is\_sync**

**Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**subcluster1.ext\_abort\_so\_write\_is\_sync**

**Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster1.has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster1.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster1.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster1.memory\_tagging\_support\_level****Type**

int

**Default value**

0x2

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

**subcluster1.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**subcluster1.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster2.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**subcluster2.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster2.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster2.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster2.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster2.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.



**subcluster2.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster2.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster2.cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**subcluster2.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster2.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**subcluster2.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**subcluster2.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster2.cpuX.l2cache-size****Type**

int

**Default value**

0x100000

**Description**

L2 Cache size in bytes.

**subcluster2.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster2.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster2.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster2.cpuX.semihosting-T32\_HLT**

**Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-Thumb\_SVC**

**Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster2.cpuX.semihosting-cmd\_line**

**Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster2.cpuX.semihosting-cwd**

**Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster2.cpuX.semihosting-enable**

**Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster2.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster2.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster2.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster2.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster2.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster2.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster2.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster2.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster2.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.



**subcluster2.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster2.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster2.error\_record\_feature\_register****Type**

string

**Default value**

```
"[{\"ED\":0x2,\"IMPDEF_3_2\":0x0,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,
\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0,\"Visibility\": \"Cluster
\"},{\"ED\":0x2,\"IMPDEF_3_2\":0x0,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,
\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0}],"
```

**Description**

RAS feature register values. An array of JSON objects. The JSON schema for the array is:

```
[{"ED":0x0, "IMPDEF_3_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0,
"DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0,
"UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility": "Core"}, other_feature_register_values].
```

Where ED, UI, FI, CE and UE have valid values between 0x0 - 0x3. CFI and

DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0, 0x2 or 0x4.

RP, CEO, INJ, FRX, UC, UEU, UER, UEO, DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

**subcluster2.ete.CLAIMTAGS****Type**

int

**Default value**

0x20

**Description**

Number of claim tags.

**subcluster2.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster2.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster2.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster2.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster2.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster2.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster2.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster2.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster2.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster2.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster2.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster2.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster2.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster2.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster2.ext\_abort\_normal\_noncacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of normal noncacheable-read external aborts.

**subcluster2.ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x2

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**subcluster2.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster2.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster2.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster2.has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster2.has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**subcluster2.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster2.has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**subcluster2.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster2.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster2.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster2.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster2.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster2.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.



**subcluster2.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**subcluster2.instruction\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**subcluster2.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster2.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster2.pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**subcluster2.pseudo\_fault\_generation\_feature\_register****Type**

string

**Default value**

```
"[{\"OF\":true, \"UC\":true, \"UEU\":false, \"UER\":false, \"UEO\":false, \"DE\":0x1, \"CE\":0x1,
\"CI\":true, \"ER\":false, \"PN\":true, \"AV\":false, \"MV\":true, \"SYN\":true, \"R\":true},{\"OF
\":false, \"UC\":true, \"UEU\":false, \"UER\":false, \"UEO\":false, \"DE\":0x1, \"CE\":0x1, \"CI
\":false, \"ER\":false, \"PN\":false, \"AV\":false, \"MV\":false, \"SYN\":false, \"R\":true}]"
```

**Description**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_psuedo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

**subcluster2.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**subcluster2.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster2.trace\_physical\_registers\_when\_host\_virtualisation\_enabled****Type**

int

**Default value**

0x1

**Description**

When host virtualisation is enabled, trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers.

**subcluster2.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster2.walk\_cache\_latency****Type**

int

**Default value**

0x0

Description

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.38 ARMCortexA510CT\_CortexA715CT\_CortexX3CT

ARMCortexA510CT\_CortexA715CT\_CortexX3CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-331: IP revisions support

Revision	Quality level
CortexA510 r0p0	Full support
CortexA510 r1p3	Full support
CortexA715 r1p2	Full support
CortexX3 r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexA510CT\_CortexA715CT\_CortexX3CT

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**  
Possible values are 1-10 (ARMCortexA510CT).

**subcluster1.NUM\_CORES**  
Possible values are 1-10 (ARMCortexA715CT).

**subcluster2.NUM\_CORES**  
Possible values are 1-10 (ARMCortexX3CT).

The total number of cores in the cluster cannot exceed 12.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-9] for cores in subcluster0.
- <port\_name>[10-19] for cores in subcluster1.
- <port\_name>[20-29] for cores in subcluster2.



All instances in the Master cross trigger matrix port array, cti[30] must be connected, regardless of the NUM\_CORES value used.

Core-specific parameters have the following prefixes:

- subcluster0.cpu0 to subcluster0.cpu9 identify cores in subcluster0.

- subcluster1.cpu0 to subcluster1.cpu9 identify cores in subcluster1.
- subcluster2.cpu0 to subcluster2.cpu9 identify cores in subcluster2.

For information about the cores in this model, see:

- [ARMCortexA510CT](#).
- [ARMCortexA715CT](#).
- [ARMCortexX3CT](#).

## Iris and MTI instances for ARMCortexA510CT\_CortexA715CT\_CortexX3CT

This model has the following Iris instances:

**Table 3-332: ARMCortexA510CT\_CortexA715CT\_CortexX3CT Iris instances**

InstanceName	ComponentName
ARMCortexA510CT_CortexA715CT_CortexX3CT	Cluster_ARM_CortexA510_CortexA715_CortexX3_Heterogeneous
ARMCortexA510CT_CortexA715CT_CortexX3CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU	DSU
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core1	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core2	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core2.busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[3]	PVBusSlave

InstanceName	ComponentName
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[7]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.cpu0.debug_rom	debug_rom
ARMCortexA510CT_CortexA715CT_CortexX3CT.cpu1.debug_rom	debug_rom
ARMCortexA510CT_CortexA715CT_CortexX3CT.cpu2.debug_rom	debug_rom
ARMCortexA510CT_CortexA715CT_CortexX3CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA715CT_CortexX3CT.global_debug_rom	debug_rom
ARMCortexA510CT_CortexA715CT_CortexX3CT.secondary_debug_rom	debug_rom
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0	Subcluster_ARM_Cortex-A510
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.dtlb	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave

InstanceName	ComponentName
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.sve	ScalableVectorExtension
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1	Subcluster_ARM_Cortex-A715
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0	ARM_Cortex-A715
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu1.dtlb	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.sve	ScalableVectorExtension
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2	Subcluster_ARM_Cortex-X3
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0	ARM_Cortex-X3
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l1licache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu2.dtlb	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-333: ARMCortexA510CT\_CortexA715CT\_CortexX3CT MTI instances**

InstanceName	ComponentName
ARMCortexA510CT_CortexA715CT_CortexX3CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU	DSU
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core1	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core2	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_core2.busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[7]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.gic_cpuif_decoder_cluster	GiCv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l1icache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l1icache.upstream[0]	PVBusSlave



InstanceName	ComponentName
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0	ARM_Cortex-A715
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l1icache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0	ARM_Cortex-X3
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.UTLB	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l1dcache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l1icache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l2cache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0.l2cache.upstream[1]	PVBusSlave

### Ports for ARMCortexA510CT\_CortexA715CT\_CortexX3CT

**Table 3-334: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).

Name	Protocol	Type	Description
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[30]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[30]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[30]	Signal	Master	Timer signals to SOC
CNTHVIRQ[30]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[30]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[30]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[30]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[30]	Signal	Master	Timer signals to SOC.
commirq[30]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[30]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel[30]	PChannel	Master	Core PCSM signals

Name	Protocol	Type	Description
core_powerdown_out[30]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[30]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[30]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[30]	Signal	Slave	Disable cryptography extensions after reset.
cti[30]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[30]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[30]	Signal	Master	No power-down request.
dbgpwrupreq[30]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[30]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[30]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[30]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[30]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[30]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[30]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[30]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.

Name	Protocol	Type	Description
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[30]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[30]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[30]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[30]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[30]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[30]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[30]	Signal	Slave	Virtualised FIQ.
virq[30]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[30]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA510CT\_CortexA715CT\_CortexX3CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

**Description**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

**AEND2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**default\_opmode****Type**

int

**Default value**

0x4



**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x0

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate

the mpm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantums in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**subcluster0.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.



**subcluster0.cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x20

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**subcluster0.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster0.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster0.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster0.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster0.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.



**subcluster0.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster0.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster0.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster0.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster0.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster0.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster0.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster0.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster0.has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster0.max\_32bit\_el****Type**

int

**Default value**

0x0

**Description**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: ELO supports A32. This parameter is ignored in Rev0 i.e. when revision\_number = 0.

**subcluster0.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**subcluster0.revision\_number****Type**

int

**Default value**

0x1

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).



**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster1.cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**subcluster1.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster1.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.max_code_cache_mb`****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.



**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.error\_record\_feature\_register****Type**

string

**Default value**

```
"[{\"ED\":0x2,\"IMPDEF_3_2\":0x1,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x2,\"TS\":0x0,\"Visibility\":\"Cluster\"},{\"ED\":0x2,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0}],"
```

**Description**

RAS feature register values. An array of JSON objects. The JSON schema for the array is:  
 [{"ED":0x0, "IMPDEF\_3\_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other\_feature\_register\_values].  
 Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4.  
 RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

**subcluster1.ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster1.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster1.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster1.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster1.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster1.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster1.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster1.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster1.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster1.ete.REVISION****Type**

int

**Default value**

0x1

**Description**

TRCIDR1 revision value.

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster1.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster1.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster1.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster1.has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster1.has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**subcluster1.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster1.has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**subcluster1.has\_statistical\_profiling****Type**

bool

**Default value**

0x0

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0



**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`subcluster1.icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`subcluster1.icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster1.icache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**`subcluster1.instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster1.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster1.pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**subcluster1.pseudo\_fault\_generation\_feature\_register****Type**

string

**Default value**

"[{\"OF\":true, \"UC\":true, \"UEU\":false, \"UER\":false, \"UEO\":false, \"DE\":0x1, \"CE\":0x1, \"CI\":true, \"ER\":false, \"PN\":true, \"AV\":false, \"MV\":true, \"SYN\":true, \"R\":true},{\"OF\":false, \"UC\":true, \"UEU\":false, \"UER\":false, \"UEO\":false, \"DE\":0x1, \"CE\":0x1, \"CI\":false, \"ER\":false, \"PN\":false, \"AV\":false, \"MV\":false, \"SYN\":false, \"R\":true}]"

**Description**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false,

"CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_psuedo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

### **subcluster1.ptw\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

### **subcluster1.stage12\_tlb\_size**

#### **Type**

int

#### **Default value**

0x80

#### **Description**

Number of stage1+2 tlb entries.

### **subcluster1.tlb\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

### **subcluster1.tlbi\_stall\_enabled**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster1.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster2.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**subcluster2.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster2.cpi\_mul**

**Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster2.cpuX.CFGEND**

**Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster2.cpuX.CFGTE**

**Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster2.cpuX.CRYPTODISABLE**

**Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster2.cpuX.RVBARADDR**

**Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster2.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster2.cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**subcluster2.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-size****Type**

int

**Default value**

0x100000

**Description**

L2 Cache size in bytes.

**subcluster2.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100



**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**`subcluster2.cpuX.min_sync_level`****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**`subcluster2.cpuX.semihosting-A32_HLT`****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**`subcluster2.cpuX.semihosting-A64_HLT`****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**`subcluster2.cpuX.semihosting-ARM_SVC`****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**`subcluster2.cpuX.semihosting-T32_HLT`****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster2.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster2.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster2.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster2.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster2.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster2.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster2.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster2.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster2.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster2.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster2.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster2.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster2.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster2.error\_record\_feature\_register****Type**

string

**Default value**

"[{\"ED\":0x2,\"IMPDEF\_3\_2\":0x0,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0,\"Visibility\":\"Cluster\"},{\"ED\":0x2,\"IMPDEF\_3\_2\":0x0,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0}]]"

**Description**

RAS feature register values. An array of JSON objects. The JSON schema for the array is:  
 [{"ED":0x0, "IMPDEF\_3\_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other\_feature\_register\_values].  
 Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4.  
 RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

**subcluster2.ete.CLAIMTAGS****Type**

int

**Default value**

0x20

**Description**

Number of claim tags.

**subcluster2.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster2.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster2.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster2.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster2.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster2.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster2.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster2.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.



**subcluster2.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster2.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster2.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster2.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster2.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster2.etc.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster2.ext\_abort\_normal\_noncacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of normal noncacheable-read external aborts.

**subcluster2.ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x2

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**subcluster2.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster2.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster2.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster2.has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**subcluster2.has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**subcluster2.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster2.has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**subcluster2.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster2.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster2.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster2.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster2.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster2.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster2.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**subcluster2.instruction\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**subcluster2.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster2.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster2.pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**subcluster2.pseudo\_fault\_generation\_feature\_register****Type**

string

**Default value**

```
"[{\"OF\":true, \"UC\":true, \"UEU\":false, \"UER\":false, \"UEO\":false, \"DE\":0x1, \"CE\":0x1,
\"CI\":true, \"ER\":false, \"PN\":true, \"AV\":false, \"MV\":true, \"SYN\":true, \"R\":true},{\"OF
\":false, \"UC\":true, \"UEU\":false, \"UER\":false, \"UEO\":false, \"DE\":0x1, \"CE\":0x1, \"CI
\":false, \"ER\":false, \"PN\":false, \"AV\":false, \"MV\":false, \"SYN\":false, \"R\":true}]"
```

**Description**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_pseudo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

**subcluster2.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**subcluster2.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster2.trace\_physical\_registers\_when\_host\_virtualisation\_enabled****Type**

int

**Default value**

0x1

**Description**

When host virtualisation is enabled, trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers.

**subcluster2.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster2.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.



### 3.5.39 ARM Cortex A520 AECT

ARM Cortex A520 AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-335: IP revisions support**

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Changes in 11.29.19

Parameters added:

- `has_coherent_icache`

#### AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.
- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.

As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

#### Iris and MTI instances for ARM Cortex A520 AECT

This model has the following Iris instances:

**Table 3-336: ARM Cortex A520 AECT Iris instances**

InstanceName	ComponentName
ARM Cortex A520 AECT	Cluster_ARM_Cortex-A520AE
ARM Cortex A520 AECT.AMU	PVBusLogger
ARM Cortex A520 AECT.AMU.mapper	PVBusMapper
ARM Cortex A520 AECT.DAP	PVBusLogger
ARM Cortex A520 AECT.DAP.mapper	PVBusMapper
ARM Cortex A520 AECT.DSU	DSU
ARM Cortex A520 AECT.DSU.PPU_cluster	PPUv1
ARM Cortex A520 AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARM Cortex A520 AECT.DSU.PPU_core0	PPUv1

InstanceName	ComponentName
ARMCortexA520AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache	PVCache
ARMCortexA520AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA520AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520AECT.MMAP	PVBusLogger
ARMCortexA520AECT.MMAP.mapper	PVBusMapper
ARMCortexA520AECT.RAS	PVBusLogger
ARMCortexA520AECT.RAS.mapper	PVBusMapper
ARMCortexA520AECT.cpu0	ARM_Cortex-A520AE
ARMCortexA520AECT.cpu0.UTLB	TLB
ARMCortexA520AECT.cpu0.debug_rom	debug_rom
ARMCortexA520AECT.cpu0.dtlb	TLB
ARMCortexA520AECT.cpu0.l1dcache	PVCache
ARMCortexA520AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520AECT.cpu0.l1icache	PVCache
ARMCortexA520AECT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520AECT.cpu0.l2cache	PVCache
ARMCortexA520AECT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520AECT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520AECT.ext_bus	PVBusLogger
ARMCortexA520AECT.ext_bus.mapper	PVBusMapper
ARMCortexA520AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520AECT.global_debug_rom	debug_rom
ARMCortexA520AECT.secondary_debug_rom	debug_rom
ARMCortexA520AECT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-337: ARMCortexA520AECT MTI instances**

InstanceName	ComponentName
ARMCortexA520AECT.AMU	PVBusLogger
ARMCortexA520AECT.AMU.mapper	PVBusMapper
ARMCortexA520AECT.DAP	PVBusLogger
ARMCortexA520AECT.DAP.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA520AECT.DSU	DSU
ARMCortexA520AECT.DSU.PPU_cluster	PPUv1
ARMCortexA520AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520AECT.DSU.PPU_core0	PPUv1
ARMCortexA520AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache	PVCache
ARMCortexA520AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA520AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520AECT.MMAP	PVBusLogger
ARMCortexA520AECT.MMAP.mapper	PVBusMapper
ARMCortexA520AECT.RAS	PVBusLogger
ARMCortexA520AECT.RAS.mapper	PVBusMapper
ARMCortexA520AECT.cpu0	ARM_Cortex-A520AE
ARMCortexA520AECT.cpu0.UTLB	TLB
ARMCortexA520AECT.cpu0.l1dcache	PVCache
ARMCortexA520AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520AECT.cpu0.l1icache	PVCache
ARMCortexA520AECT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520AECT.cpu0.l2cache	PVCache
ARMCortexA520AECT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520AECT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520AECT.ext_bus	PVBusLogger
ARMCortexA520AECT.ext_bus.mapper	PVBusMapper
ARMCortexA520AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA520AECT

**Table 3-338: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).

Name	Protocol	Type	Description
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[14]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsmpchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[14]	Signal	Master	Timer signals to SOC
CNTHVIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[14]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[14]	Signal	Master	Timer signals to SOC.

Name	Protocol	Type	Description
CNTPSIRQ[14]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[14]	Signal	Master	Timer signals to SOC.
commirq[14]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[14]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[14]	PChannel	Master	Core PCSM signals
core_powerdown_out[14]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[14]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[14]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[14]	Signal	Slave	Disable cryptography extensions after reset.
cti[14]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[14]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[14]	Signal	Master	No power-down request.
dbgprupreq[14]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[14]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[14]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[14]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmuirq[14]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.

Name	Protocol	Type	Description
ppu_core_irq[14]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[14]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[14]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[14]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[14]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[14]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[14]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[14]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[14]	Signal	Slave	Virtualised FIQ.
virq[14]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[14]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A520 AECT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

**AEND1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

**AEND2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0



**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x0

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**cluster\_split\_lock\_config****Type**

int

**Default value**

0x1

**Description**

Default SPLIT/LOCKED config. Directly maps to values of CLUSTERSLCFR. The valid values are: 1 - Only LOCKED configuration support 4 - Only SPLIT configuration support, 5 - Mixed Configuration support. Modes are not software visible, and not modeled. Valid only when enable\_ae\_features is true.

**core\_cache\_protection****Type**

int

**Default value**

0x1

**Description**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

**core\_complex\_mapping****Type**

string

**Default value**

```
"{"complex0": { "cores": [0, 1], "l2-cache" :{"exists":1, "size":"16MB"}}, "complex1":
{ "cores": [2, 3], "l2-cache" :{"exists":1, "size":"16MB"}}, "complex2": { "cores": [4,
5], "l2-cache" :{"exists":1, "size":"16MB"}}, "complex3": { "cores": [6, 7], "l2-cache
" :{"exists":1, "size":"16MB"}}, "complex4": { "cores": [8, 9], "l2-cache" :{"exists
":1, "size":"16MB"}}, "complex5": { "cores": [10, 11], "l2-cache" :{"exists":1, "size":
"16MB"}}, "complex6": { "cores": [12, 13], "l2-cache" :{"exists":1, "size":"16MB"} }"
```

**Description**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" :{"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>], "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_bus_width_in_bytes`****Type**

int

**Default value**

0x10

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x20

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100



**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**enable\_lock\_step****Type**

bool

**Default value**

0x0

**Description**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CMOD value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3



**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**has\_actlr2****Type**

bool

**Default value**

0x1

**Description**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

**has\_coherent\_icache****Type**

bool

**Default value**

0x0

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_impdef\_transient\_fault\_protection****Type**

bool

**Default value**

0x1

**Description**

Support the Transient Fault Protection (TFP) flop parity errors through RAS registers (FEAT\_TFP).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l3cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied.

This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int



**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of  $n$  means the accumulator will use  $(n * \text{accumulator value})$  to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_acp****Type**

int

**Default value**

0x0

**Description**

Number of ACP ports.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**ras\_extra\_configurations****Type**

string

**Default value**

```
"[ { \"Index\": 1, \"ERXMISCO_mask\": 0xFFFFc0003fc3, \"ERXMISC1_mask\":  
0x03F870003FF30f07, \"ERXPFGCTL_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISCO_mask  
\": 0xFFFFe007ffc0, \"ERXMISCO_reset\": 0x2, \"ERXSTATUS_IERR_mask\": 0x300 ,  
\"ERXMISC1_mask\": 0x0FF8700ffff31f0f, \"ERXPFGCTL_reset\": 0x1000 } ]"
```

**Description**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCN masks - these are 64 bit masks covering the 64 bit registers ERXMISCN\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

**ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.40 ARMCortexA520CT

ARMCortexA520CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-339: IP revisions support**

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About ARMCortexA520CT

The model supports the following features:

- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- Internal PPU support is present.
- A P-Channel for the cluster and for each core.
- BROADCASTPERSIST pin.
- Optional peripheral port.
- Memory-mapped register access to MPAM.
- Per-core clock.
- Utility bus.

Support for the following features is planned for a future release:

- DSU-120 system features are not fully implemented.
- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (TLM2 extensions from Arm) bus protocols.
- Core-Complex.
- BROADCASTCACHEMAINTPOU pin
- COREINSTRRET and COREINSTRRUN signals.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not supported but signals are implemented.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
- Automatic CPU retention mode.
- Level-3 Cache RAM retention.

- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARM Cortex-A520CT

This model has the following Iris instances:

**Table 3-340: ARM Cortex-A520CT Iris instances**

InstanceName	ComponentName
ARM Cortex-A520CT	Cluster_ARM_Cortex-A520
ARM Cortex-A520CT.AMU	PVBusLogger
ARM Cortex-A520CT.AMU.mapper	PVBusMapper
ARM Cortex-A520CT.DAP	PVBusLogger
ARM Cortex-A520CT.DAP.mapper	PVBusMapper
ARM Cortex-A520CT.DSU	DSU
ARM Cortex-A520CT.DSU.PPU_cluster	PPUv1
ARM Cortex-A520CT.DSU.PPU_cluster.busslave	PVBusSlave
ARM Cortex-A520CT.DSU.PPU_core0	PPUv1
ARM Cortex-A520CT.DSU.PPU_core0.busslave	PVBusSlave
ARM Cortex-A520CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM Cortex-A520CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex-A520CT.DSU.shared_cache	PVCache
ARM Cortex-A520CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM Cortex-A520CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM Cortex-A520CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM Cortex-A520CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARM Cortex-A520CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARM Cortex-A520CT.DSU.utility_slave[0]	PVBusSlave
ARM Cortex-A520CT.MMAP	PVBusLogger
ARM Cortex-A520CT.MMAP.mapper	PVBusMapper
ARM Cortex-A520CT.RAS	PVBusLogger
ARM Cortex-A520CT.RAS.mapper	PVBusMapper
ARM Cortex-A520CT.cpu0	ARM_Cortex-A520

InstanceName	ComponentName
ARMCortexA520CT.cpu0.UTLB	TLB
ARMCortexA520CT.cpu0.debug_rom	debug_rom
ARMCortexA520CT.cpu0.dtlb	TLB
ARMCortexA520CT.cpu0.l1dcache	PVCache
ARMCortexA520CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT.cpu0.l1icache	PVCache
ARMCortexA520CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520CT.cpu0.l2cache	PVCache
ARMCortexA520CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT.ext_bus	PVBusLogger
ARMCortexA520CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT.global_debug_rom	debug_rom
ARMCortexA520CT.secondary_debug_rom	debug_rom
ARMCortexA520CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-341: ARMCortexA520CT MTI instances**

InstanceName	ComponentName
ARMCortexA520CT.AMU	PVBusLogger
ARMCortexA520CT.AMU.mapper	PVBusMapper
ARMCortexA520CT.DAP	PVBusLogger
ARMCortexA520CT.DAP.mapper	PVBusMapper
ARMCortexA520CT.DSU	DSU
ARMCortexA520CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT.DSU.PPU_core0	PPUv1
ARMCortexA520CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT.DSU.shared_cache	PVCache
ARMCortexA520CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA520CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA520CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA520CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA520CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA520CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT.MMAP	PVBusLogger

InstanceName	ComponentName
ARMCortexA520CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT.RAS	PVBusLogger
ARMCortexA520CT.RAS.mapper	PVBusMapper
ARMCortexA520CT.cpu0	ARM_Cortex-A520
ARMCortexA520CT.cpu0.UTLB	TLB
ARMCortexA520CT.cpu0.l1dcache	PVCache
ARMCortexA520CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT.cpu0.l1icache	PVCache
ARMCortexA520CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520CT.cpu0.l2cache	PVCache
ARMCortexA520CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT.ext_bus	PVBusLogger
ARMCortexA520CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA520CT

**Table 3-342: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[14]	Signal	Slave	This signal is for EE bit initialisation.



Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamIQ pmu irq
CNTHPIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[14]	Signal	Master	Timer signals to SOC
CNTHVIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[14]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[14]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[14]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[14]	Signal	Master	Timer signals to SOC.
commirq[14]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[14]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel[14]	PChannel	Master	Core PCSM signals
core_powerdown_out[14]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[14]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[14]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[14]	Signal	Slave	Disable cryptography extensions after reset.

Name	Protocol	Type	Description
cti[14]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[14]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[14]	Signal	Master	No power-down request.
dbgpwrupreq[14]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[14]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[14]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[14]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmuirq[14]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[14]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[14]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[14]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[14]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[14]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.

Name	Protocol	Type	Description
ticks[14]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[14]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[14]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[14]	Signal	Slave	Virtualised FIQ.
virq[14]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[14]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA520CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAIN****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x0

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_cache\_protection****Type**

int

**Default value**

0x1

**Description**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

**core\_complex\_mapping****Type**

string

**Default value**

```
"{\complex0\": { \"cores\": [0, 1], \"l2-cache\" :{\"exists\":1, \"size\":\"16MB\"}}, \"complex1\":  
{ \"cores\": [2, 3], \"l2-cache\" :{\"exists\":1, \"size\":\"16MB\"}}, \"complex2\": { \"cores\": [4,  
5], \"l2-cache\" :{\"exists\":1, \"size\":\"16MB\"}}, \"complex3\": { \"cores\": [6, 7], \"l2-cache  
\" :{\"exists\":1, \"size\":\"16MB\"}}, \"complex4\": { \"cores\": [8, 9], \"l2-cache\" :{\"exists  
\":1, \"size\":\"16MB\"}}, \"complex5\": { \"cores\": [10, 11], \"l2-cache\" :{\"exists\":1, \"size\":  
\"16MB\"}}, \"complex6\": { \"cores\": [12, 13], \"l2-cache\" :{\"exists\":1, \"size\":\"16MB\"}} }
```

**Description**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" :{"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>], "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int



**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CRYPTODISABLE`****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**`cpuX.RVBARADDR`****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**`cpuX.enable_trace_special_hlt_imm16`****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-

read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x20

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0



**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**`dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect.  
1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**has\_actlr2****Type**

bool

**Default value**

0x1

**Description**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.



**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks.  
This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l3cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_acp****Type**

int

**Default value**

0x0

**Description**

Number of ACP ports.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**ras\_extra\_configurations****Type**

string

**Default value**

```
"[ { \"Index\": 1, \"ERXMISC0_mask\": 0xFFFFc0003fc3, \"ERXMISC1_mask\":  
0x03F870003FF30f07, \"ERXPFGCTL_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISC0_mask  
\": 0xFFFFe007ffc0, \"ERXMISC0_reset\": 0x2, \"ERXSTATUS_IERR_mask\": 0x300 ,  
\"ERXMISC1_mask\": 0x0FF8700ffff31f0f, \"ERXPFGCTL_reset\": 0x1000 } ]"
```

**Description**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCN masks - these are 64 bit masks covering the 64 bit registers ERXMISCN\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

**ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**revision\_number****Type**

int

**Default value**

0x0

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**`treat_PAC_as_NOP`**

**Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**`walk_cache_latency`**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.41 ARMCortexA520CT\_CortexA720CT

ARMCortexA520CT\_CortexA720CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-343: IP revisions support

Revision	Quality level
CortexA520 r0p1	Full support
CortexA720 r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexA520CT\_CortexA720CT

The number of cores in each subcluster is configurable using the following parameters:

**`subcluster0.NUM_CORES`**

Possible values are 1-13 (ARMCortexA520CT).

**`subcluster1.NUM_CORES`**

Possible values are 1-13 (ARMCortexA720CT).



The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- `<port_name>[0-12]` for cores in `subcluster0`.
- `<port_name>[13-25]` for cores in `subcluster1`.



All instances in the Master cross trigger matrix port array, `cti[26]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu12` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu12` identify cores in `subcluster1`.

For information about the cores in this model, see:

- [ARMCortexA520CT](#).
- [ARMCortexA720CT](#).

## Iris and MTI instances for ARMCortexA520CT\_CortexA720CT

This model has the following Iris instances:

**Table 3-344: ARMCortexA520CT\_CortexA720CT Iris instances**

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT	Cluster_ARM_CortexA520_CortexA720_Heterogeneous
ARMCortexA520CT_CortexA720CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA720CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA720CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.DSU	DSU
ARMCortexA520CT_CortexA720CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA720CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.PPU_core0	PPUv1
ARMCortexA520CT_CortexA720CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.PPU_core1	PPUv1
ARMCortexA520CT_CortexA720CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA720CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA720CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA720CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.cpu0.debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT.cpu1.debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA720CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA720CT.global_debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT.secondary_debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT.subcluster0	Subcluster_ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.dtlb	TLB
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster0.sve	ScalableVectorExtension
ARMCortexA520CT_CortexA720CT.subcluster1	Subcluster_ARM_Cortex-A720
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0	ARM_Cortex-A720
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l1licache	PVCache

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster1.cpu1.dtlb	TLB
ARMCortexA520CT_CortexA720CT.subcluster1.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-345: ARMCortexA520CT\_CortexA720CT MTI instances**

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA720CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA720CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.DSU	DSU
ARMCortexA520CT_CortexA720CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA720CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.PPU_core0	PPUv1
ARMCortexA520CT_CortexA720CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.PPU_core1	PPUv1
ARMCortexA520CT_CortexA720CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA720CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA720CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA720CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.ext_bus	PVBusLogger

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0	ARM_Cortex-A720
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave

## Ports for ARMCortexA520CT\_CortexA720CT

**Table 3-346: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.

Name	Protocol	Type	Description
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[26]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[26]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[26]	Signal	Master	Timer signals to SOC
CNTHVIRQ[26]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[26]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[26]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[26]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[26]	Signal	Master	Timer signals to SOC.
commirq[26]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[26]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel[26]	PChannel	Master	Core PCSM signals
core_powerdown_out[26]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info

Name	Protocol	Type	Description
coreerrirq[26]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[26]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[26]	Signal	Slave	Disable cryptography extensions after reset.
cti[26]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[26]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[26]	Signal	Master	No power-down request.
dbgpwrupreq[26]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[26]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[26]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[26]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[26]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[26]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[26]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[26]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[26]	Signal	Slave	Per core RAM Error Interrupt.

Name	Protocol	Type	Description
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[26]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[26]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[26]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[26]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[26]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[26]	Signal	Slave	Virtualised FIQ.
virq[26]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[26]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM CortexA520CT\_CortexA720CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

**AEND2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int



**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x0

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_complex\_mapping****Type**

string

**Default value**

"{\\"complex0\\": { \\"cores\\": [0, 1], \\"l2-cache\\": {\\"exists\\":1, \\"size\\":\\"16MB\\"}}, \\"complex1\\": { \\"cores\\": [2, 3], \\"l2-cache\\": {\\"exists\\":1, \\"size\\":\\"16MB\\"}}, \\"complex2\\": { \\"cores\\": [4,

```
5], \"l2-cache\": { \"exists\": 1, \"size\": \"16MB\" }, \"complex3\": { \"cores\": [6, 7], \"l2-cache\": { \"exists\": 1, \"size\": \"16MB\" }, \"complex4\": { \"cores\": [8, 9], \"l2-cache\": { \"exists\": 1, \"size\": \"16MB\" }, \"complex5\": { \"cores\": [10, 11], \"l2-cache\": { \"exists\": 1, \"size\": \"16MB\" }, \"complex6\": { \"cores\": [12, 13], \"l2-cache\": { \"exists\": 1, \"size\": \"16MB\" } }
```

### Description

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {\"complex0\": { \"cores\" : [ 0, 1 ], \"l2-cache\" : { \"exists\": 1, \"size\": 16MB }, ... , \"complexN\": { \"cores\" : [<core\_list>], \"l2-cache\" : { \"exists\": 1, \"size\": 16MB } } } where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

### core\_power\_on\_by\_default

#### Type

bool

#### Default value

0x0

#### Description

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

### dcache-state\_modelled

#### Type

bool

#### Default value

0x0

#### Description

Set whether D-cache has stateful implementation.

### enable\_simulation\_performance\_optimizations

#### Type

bool

#### Default value

0x1

#### Description

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x0

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied.

This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int



**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**revision\_number****Type**

int

**Default value**

0x0

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**subcluster0.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster0.core\_cache\_protection****Type**

int

**Default value**

0x1

**Description**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x20

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC**

**Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT**

**Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-Thumb\_SVC**

**Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line**

**Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd**

**Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000



**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster0.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster0.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster0.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster0.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster0.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster0.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster0.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster0.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster0.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster0.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster0.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster0.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster0.has\_actlr2****Type**

bool

**Default value**

0x1

**Description**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster0.icache-miss\_latency****Type**

int



**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster0.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster0.pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.ras\_extra\_configurations****Type**

string

**Default value**

```
"[ { \"Index\": 1, \"ERXMISCO_mask\": 0xFFFFc0003fc3, \"ERXMISC1_mask\":  
0x03F870003FF30f07, \"ERXPFGCTL_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISCO_mask  
\": 0xFFFFe007ffc0, \"ERXMISCO_reset\": 0x2, \"ERXSTATUS_IERR_mask\": 0x300 ,  
\"ERXMISC1_mask\": 0x0FF8700ffff31f0f, \"ERXPFGCTL_reset\": 0x1000 } ]"
```

**Description**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCn masks - these are 64 bit masks covering the 64 bit registers ERXMISCn\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

**subcluster0.ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**subcluster0.revision\_number****Type**

int

**Default value**

0x0

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster1.cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**subcluster1.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster1.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.



**subcluster1.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC**

**Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT**

**Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC**

**Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line**

**Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd**

**Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster1.ete.ETE\_REVISION****Type**

int

**Default value**

0x1

**Description**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

**subcluster1.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster1.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster1.etc.PIDR\_CM0D**

**Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster1.etc.PIDR\_REVAND**

**Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster1.etc.PIDR\_REVISION**

**Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster1.etc.Q\_CADENCE**

**Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster1.etc.RES0\_STATEFUL**

**Type**

bool

**Default value**

0x0



**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster1.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster1.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster1.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster1.etc.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster1.etc.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster1.etc.TSMARK****Type**

bool

**Default value**

0x1

**Description**

Whether timestamp markers are supported.

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster1.has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**subcluster1.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster1.has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**subcluster1.has\_statistical\_profiling****Type**

bool

**Default value**

0x0

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**subcluster1.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**subcluster1.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**subcluster1.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**subcluster1.instruction\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster1.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster1.pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**subcluster1.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**subcluster1.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

Description

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

subcluster1.walk\_cache\_latency

Type

int

Default value

0x0

Description

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.42 ARMCortexA520CT\_CortexA720CT\_CortexX4CT

ARMCortexA520CT\_CortexA720CT\_CortexX4CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-347: IP revisions support

Revision	Quality level
CortexA520 r0p1	Full support
CortexA720 r0p1	Full support
CortexX4 r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexA520CT\_CortexA720CT\_CortexX4CT

The number of cores in each subcluster is configurable using the following parameters:

subcluster0.NUM\_CORES

Possible values are 1-12 (ARMCortexA520CT).

subcluster1.NUM\_CORES

Possible values are 1-12 (ARMCortexA720CT).

subcluster2.NUM\_CORES

Possible values are 1-12 (ARMCortexX4CT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-11] for cores in subcluster0.



- `<port_name>`[12-23] for cores in `subcluster1`.
- `<port_name>`[24-35] for cores in `subcluster2`.



All instances in the Master cross trigger matrix port array, `cti[36]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu11` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu11` identify cores in `subcluster1`.
- `subcluster2.cpu0` to `subcluster2.cpu11` identify cores in `subcluster2`.

For information about the cores in this model, see:

- [ARMCortexA520CT](#).
- [ARMCortexA720CT](#).
- [ARMCortexX4CT](#).

### Iris and MTI instances for ARMCortexA520CT\_CortexA720CT\_CortexX4CT

This model has the following Iris instances:

**Table 3-348: ARMCortexA520CT\_CortexA720CT\_CortexX4CT Iris instances**

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT_CortexX4CT	Cluster_ARM_CortexA520_CortexA720_CortexX4_Heterogeneous
ARMCortexA520CT_CortexA720CT_CortexX4CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU	DSU
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core0	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core1	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core2	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core2.busslave	PVBusSlave

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[7]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[8]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.cpu0.debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT_CortexX4CT.cpu1.debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT_CortexX4CT.cpu2.debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT_CortexX4CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA720CT_CortexX4CT.global_debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT_CortexX4CT.secondary_debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0	Subcluster_ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.dtlb	TLB

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.sve	ScalableVectorExtension
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1	Subcluster_ARM_Cortex-A720
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0	ARM_Cortex-A720
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu1.dtlb	TLB
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.sve	ScalableVectorExtension
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2	Subcluster_ARM_Cortex-X4
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0	ARM_Cortex-X4
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l1dcache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l1icache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu2.dtlb	TLB
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-349: ARMCortexA520CT\_CortexA720CT\_CortexX4CT MTI instances**

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT_CortexX4CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU	DSU
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core0	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core1	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core2	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_core2.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[7]	PVBusSlave

InstanceName	ComponentName
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[8]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0	ARM_Cortex-A720
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0	ARM_Cortex-X4
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0.l2cache.upstream[1]	PVBusSlave

## Ports for ARMCortexA520CT\_CortexA720CT\_CortexX4CT

Table 3-350: Ports

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[36]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[36]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[36]	Signal	Master	Timer signals to SOC
CNTHVIRQ[36]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[36]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[36]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[36]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[36]	Signal	Master	Timer signals to SOC.
commirq[36]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[36]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[36]	PChannel	Master	Core PCSM signals
core_powerdown_out[36]	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
coreerrirq[36]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[36]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[36]	Signal	Slave	Disable cryptography extensions after reset.
cti[36]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[36]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[36]	Signal	Master	No power-down request.
dbgpwrupreq[36]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.

Name	Protocol	Type	Description
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[36]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[36]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[36]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[36]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[36]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[36]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[36]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[36]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[36]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[36]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[36]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[36]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[36]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.



Name	Protocol	Type	Description
vfiq[36]	Signal	Slave	Virtualised FIQ.
virq[36]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[36]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM CortexA520CT\_CortexA720CT\_CortexX4CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

### AEND3\_DEFAULT

#### Type

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x0

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_complex\_mapping****Type**

string

**Default value**

```
"{"complex0": { "cores": [0, 1], "l2-cache" : {"exists":1, "size":"16MB"}}, "complex1":
{ "cores": [2, 3], "l2-cache" : {"exists":1, "size":"16MB"}}, "complex2": { "cores": [4,
5], "l2-cache" : {"exists":1, "size":"16MB"}}, "complex3": { "cores": [6, 7], "l2-cache
" : {"exists":1, "size":"16MB"}}, "complex4": { "cores": [8, 9], "l2-cache" : {"exists
":1, "size":"16MB"}}, "complex5": { "cores": [10, 11], "l2-cache" : {"exists":1, "size":
"16MB"}}, "complex6": { "cores": [12, 13], "l2-cache" : {"exists":1, "size":"16MB"}}}"
```

**Description**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" : {"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>], "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0



**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**l3cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**revision\_number****Type**

int

**Default value**

0x0

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**subcluster0.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster0.core\_cache\_protection****Type**

int

**Default value**

0x1

**Description**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`subcluster0.cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster0.cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster0.cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster0.cpuX.l2cache-read_bus_width_in_bytes`****Type**

int

**Default value**

0x10

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x20

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT****Type**

int



**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster0.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.etc.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster0.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster0.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster0.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster0.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster0.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster0.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster0.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster0.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster0.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster0.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster0.ete.TRCSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRTA value for a forcibly traced exception.

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0



**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster0.has\_actlr2****Type**

bool

**Default value**

0x1

**Description**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`subcluster0.icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster0.icache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**`subcluster0.invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**`subcluster0.memory_tagging_support_level`****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster0.pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.ras\_extra\_configurations****Type**

string

**Default value**

```
"[ { \"Index\": 1, \"ERXMISCO_mask\": 0xFFFFc0003fc3, \"ERXMISC1_mask\": 0x03F870003FF30f07, \"ERXPFGCTL_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISCO_mask\": 0xFFFFe007ffc0, \"ERXMISCO_reset\": 0x2, \"ERXSTATUS_IERR_mask\": 0x300, \"ERXMISC1_mask\": 0x0FF8700ffff31f0f, \"ERXPFGCTL_reset\": 0x1000 } ]"
```

**Description**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCn masks - these are 64 bit masks covering the 64 bit registers ERXMISCn\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

**subcluster0.ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**subcluster0.revision\_number****Type**

int

**Default value**

0x0

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster1.cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**subcluster1.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0



**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster1.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster1.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.max_code_cache_mb`****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.etc.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.



**subcluster1.ete.ETE\_REVISION****Type**

int

**Default value**

0x1

**Description**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

**subcluster1.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster1.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster1.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster1.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster1.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster1.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster1.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster1.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster1.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster1.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster1.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster1.ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.

**subcluster1.ete.TSMARK****Type**

bool

**Default value**

0x1

**Description**

Whether timestamp markers are supported.

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster1.has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3). Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**subcluster1.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster1.has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**subcluster1.has\_statistical\_profiling****Type**

bool

**Default value**

0x0

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**`subcluster1.icache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`subcluster1.icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`subcluster1.icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`subcluster1.icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`subcluster1.icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster1.icache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**`subcluster1.instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**`subcluster1.invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster1.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster1.pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**subcluster1.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.



**subcluster1.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster1.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster2.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster2.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster2.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster2.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster2.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster2.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster2.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster2.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster2.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster2.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**subcluster2.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster2.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster2.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster2.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster2.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster2.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster2.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster2.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster2.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster2.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.



**subcluster2.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster2.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster2.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster2.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster2.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit

or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

#### **`subcluster2.dcache-read_latency`**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

#### **`subcluster2.dcache-size`**

##### **Type**

int

##### **Default value**

0x10000

##### **Description**

L1 D-Cache size in bytes.

#### **`subcluster2.dcache-snoop_data_transfer_latency`**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

#### **`subcluster2.dcache-write_access_latency`**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

#### **subcluster2.dcache-write\_latency**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

#### **subcluster2.ecv\_support\_level**

##### **Type**

int

##### **Default value**

0x2

##### **Description**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

#### **subcluster2.ete.CLAIMTAGS**

##### **Type**

int

##### **Default value**

0x4

##### **Description**

Number of claim tags.

#### **subcluster2.ete.MAX\_INST\_PER\_Q**

##### **Type**

int

##### **Default value**

0x1

##### **Description**

Maximum limit for the number of instructions implied by a Q element.

#### **subcluster2.ete.NumberOfRSPairs**

##### **Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster2.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster2.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster2.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster2.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster2.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster2.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster2.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster2.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster2.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster2.ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x2

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**subcluster2.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster2.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster2.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster2.has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**subcluster2.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster2.has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**subcluster2.has\_v8\_7\_spe\_inverted\_filtering****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

**subcluster2.has\_v8\_7\_spe\_previous\_branch\_target****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).



**subcluster2.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster2.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster2.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster2.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster2.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`subcluster2.icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster2.icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`subcluster2.instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**`subcluster2.invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster2.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster2.mpam\_has\_altsp****Type**

bool

**Default value**

0x0

**Description**

MPAM Whether MPAMIDR\_EL1.HAS\_ALTSP bit is set or clear.

**subcluster2.mpamidr\_has\_force\_ns****Type**

int

**Default value**

0x0

**Description**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**subcluster2.mpamidr\_has\_sdeflt****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**subcluster2.mpamidr\_has\_tidr****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear Possible values of this parameter are:  
- 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**subcluster2.pmu-num\_counters****Type**

int

**Default value**

0x1f

**Description**

Number of PMU counters implemented.

**subcluster2.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**subcluster2.tcr\_txsz\_undersize\_should\_fault****Type**

bool

**Default value**

0x0

**Description**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

**subcluster2.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlbi invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster2.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster2.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.43 ARMCortexA520CT\_CortexA725CT

ARMCortexA520CT\_CortexA725CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-351: IP revisions support**

Revision	Quality level
CortexA520 r0p1	Full support
CortexA725 r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About ARMCortexA520CT\_CortexA725CT

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 1-13 (ARMCortexA520CT).

**subcluster1.NUM\_CORES**

Possible values are 1-13 (ARMCortexA725CT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-12] for cores in subcluster0.
- <port\_name>[13-25] for cores in subcluster1.



All instances in the Master cross trigger matrix port array, `cti[26]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu12` identify cores in subcluster0.
- `subcluster1.cpu0` to `subcluster1.cpu12` identify cores in subcluster1.

For information about the cores in this model, see:

- [ARMCortexA520CT](#).
- [ARMCortexA725CT](#).

#### Iris and MTI instances for ARMCortexA520CT\_CortexA725CT

This model has the following Iris instances:

**Table 3-352: ARMCortexA520CT\_CortexA725CT Iris instances**

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT	Cluster_ARM_Cortex-A520_CortexA725_Heterogeneous
ARMCortexA520CT_CortexA725CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA725CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA725CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.DSU	DSU
ARMCortexA520CT_CortexA725CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA725CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.PPU_core0	PPUv1
ARMCortexA520CT_CortexA725CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.PPU_core1	PPUv1
ARMCortexA520CT_CortexA725CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA725CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA725CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA725CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.cpu0.debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT.cpu1.debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA725CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA725CT.global_debug_rom	debug_rom

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT.secondary_debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT.subcluster0	Subcluster_ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.dtlb	TLB
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster0.sve	ScalableVectorExtension
ARMCortexA520CT_CortexA725CT.subcluster1	Subcluster_ARM_Cortex-A725
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0	ARM_Cortex-A725
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster1.cpu1.dtlb	TLB
ARMCortexA520CT_CortexA725CT.subcluster1.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-353: ARMCortexA520CT\_CortexA725CT MTI instances**

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA725CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA725CT.DAP.mapper	PVBusMapper



InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT.DSU	DSU
ARMCortexA520CT_CortexA725CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA725CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.PPU_core0	PPUv1
ARMCortexA520CT_CortexA725CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.PPU_core1	PPUv1
ARMCortexA520CT_CortexA725CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA725CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA725CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA725CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA725CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l1icache	PVCache
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0	ARM_Cortex-A725
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l1icache	PVCache

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave

## Ports for ARMCortexA520CT\_CortexA725CT

**Table 3-354: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[26]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[26]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[26]	Signal	Master	Timer signals to SOC
CNTHVIRQ[26]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[26]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[26]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[26]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[26]	Signal	Master	Timer signals to SOC.
commirq[26]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[26]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[26]	PChannel	Master	Core PCSM signals
core_powerdown_out[26]	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
coreerrirq[26]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[26]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[26]	Signal	Slave	Disable cryptography extensions after reset.
cti[26]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[26]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[26]	Signal	Master	No power-down request.
dbgpwrupreq[26]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.

Name	Protocol	Type	Description
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[26]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[26]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[26]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[26]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[26]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[26]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[26]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[26]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[26]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[26]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[26]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[26]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[26]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.

Name	Protocol	Type	Description
vfiq[26]	Signal	Slave	Virtualised FIQ.
virq[26]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[26]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM CortexA520CT\_CortexA725CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

### AEND3\_DEFAULT

#### Type

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x0



**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_complex\_mapping****Type**

string

**Default value**

```
"{"complex0": { "cores": [0, 1], "l2-cache" : {"exists":1, "size":"16MB"}}, "complex1":
{ "cores": [2, 3], "l2-cache" : {"exists":1, "size":"16MB"}}, "complex2": { "cores": [4,
5], "l2-cache" : {"exists":1, "size":"16MB"}}, "complex3": { "cores": [6, 7], "l2-cache
" : {"exists":1, "size":"16MB"}}, "complex4": { "cores": [8, 9], "l2-cache" : {"exists
":1, "size":"16MB"}}, "complex5": { "cores": [10, 11], "l2-cache" : {"exists":1, "size":
"16MB"}}, "complex6": { "cores": [12, 13], "l2-cache" : {"exists":1, "size":"16MB"}}}"
```

**Description**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" : {"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>], "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**l3cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**revision\_number****Type**

int

**Default value**

0x0

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**subcluster0.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster0.core\_cache\_protection****Type**

int

**Default value**

0x1

**Description**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0



**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`subcluster0.cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster0.cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster0.cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster0.cpuX.l2cache-read_bus_width_in_bytes`****Type**

int

**Default value**

0x10

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x20

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-size`****Type**

int

**Default value**

0x8000



**Description**

L1 D-Cache size in bytes.

**subcluster0.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.etc.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster0.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster0.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster0.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster0.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster0.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster0.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster0.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster0.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster0.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster0.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster0.ete.TRCSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRTA value for a forcibly traced exception.

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster0.has\_actlr2****Type**

bool

**Default value**

0x1

**Description**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`subcluster0.icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster0.icache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**`subcluster0.invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**`subcluster0.memory_tagging_support_level`****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster0.pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.ras\_extra\_configurations****Type**

string

**Default value**

```
"[ { \"Index\": 1, \"ERXMISCO_mask\": 0xFFFFc0003fc3, \"ERXMISC1_mask\": 0x03F870003FF30f07, \"ERXPFGCTL_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISCO_mask\": 0xFFFFe007ffc0, \"ERXMISCO_reset\": 0x2, \"ERXSTATUS_IERR_mask\": 0x300, \"ERXMISC1_mask\": 0x0FF8700ffff31f0f, \"ERXPFGCTL_reset\": 0x1000 } ]"
```

**Description**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCn masks - these are 64 bit masks covering the 64 bit registers ERXMISCn\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].



**subcluster0.ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**subcluster0.revision\_number****Type**

int

**Default value**

0x0

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster1.cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**subcluster1.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-size****Type**

int

**Default value**

0x100000

**Description**

L2 Cache size in bytes.

**subcluster1.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.max_code_cache_mb`****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.



**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.etc.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster1.ete.ETE\_REVISION****Type**

int

**Default value**

0x1

**Description**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

**subcluster1.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster1.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster1.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster1.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster1.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster1.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster1.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster1.ete.RETSTACK****Type**

int

**Default value**

0x1

**Description**

Return stack depth.

**subcluster1.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster1.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster1.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster1.ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.

**subcluster1.ete.TSMARK****Type**

bool

**Default value**

0x1

**Description**

Whether timestamp markers are supported.

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.



**subcluster1.has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3). Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**subcluster1.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster1.has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**subcluster1.has\_statistical\_profiling****Type**

bool

**Default value**

0x0

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**subcluster1.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`subcluster1.icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster1.icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`subcluster1.instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**`subcluster1.invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster1.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster1.pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**subcluster1.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**subcluster1.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster1.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.44 ARMCortexA520CT\_CortexA725CT\_CortexX925CT

ARMCortexA520CT\_CortexA725CT\_CortexX925CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-355: IP revisions support**

Revision	Quality level
CortexA520 r0p1	Full support
CortexA725 r0p0	Preliminary support
CortexX925 r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About ARMCortexA520CT\_CortexA725CT\_CortexX925CT

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 1-12 (ARMCortexA520CT).

**subcluster1.NUM\_CORES**

Possible values are 1-12 (ARMCortexA725CT).

**subcluster2.NUM\_CORES**

Possible values are 1-12 (ARMCortexX925CT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-11] for cores in subcluster0.
- <port\_name>[12-23] for cores in subcluster1.
- <port\_name>[24-35] for cores in subcluster2.



All instances in the Master cross trigger matrix port array, `cti[36]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu11` identify cores in subcluster0.
- `subcluster1.cpu0` to `subcluster1.cpu11` identify cores in subcluster1.
- `subcluster2.cpu0` to `subcluster2.cpu11` identify cores in subcluster2.

For information about the cores in this model, see:

- [ARMCortexA520CT](#).
- [ARMCortexA725CT](#).
- [ARMCortexX925CT](#).

## Iris and MTI instances for ARMCortexA520CT\_CortexA725CT\_CortexX925CT

This model has the following Iris instances:

**Table 3-356: ARMCortexA520CT\_CortexA725CT\_CortexX925CT Iris instances**

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT_CortexX925CT	Cluster_ARM_Cortex-A520_CortexA725_CortexX925_Heterogeneous
ARMCortexA520CT_CortexA725CT_CortexX925CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU	DSU
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core0	PPUv1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core1	PPUv1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core2	PPUv1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core2.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[4]	PVBusSlave

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[7]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[8]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.cpu0.debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT_CortexX925CT.cpu1.debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT_CortexX925CT.cpu2.debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT_CortexX925CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA725CT_CortexX925CT.global_debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT_CortexX925CT.secondary_debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0	Subcluster_ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.dtlb	TLB
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave



InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.sve	ScalableVectorExtension
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1	Subcluster_ARM_Cortex-A725
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0	ARM_Cortex-A725
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu1.dtlb	TLB
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.sve	ScalableVectorExtension
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2	Subcluster_ARM_Cortex-X925
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0	ARM_Cortex-X925
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l2cache.upstream[1]	PVBusSlave

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu2.dtlb	TLB
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-357: ARMCortexA520CT\_CortexA725CT\_CortexX925CT MTI instances**

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT_CortexX925CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU	DSU
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core0	PPUv1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core1	PPUv1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core2	PPUv1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_core2.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[7]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[8]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.ext_bus	PVBusLogger

InstanceName	ComponentName
ARMCortexA520CT_CortexA725CT_CortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0	ARM_Cortex-A725
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0	ARM_Cortex-X925
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.UTLB	TLB
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l1dcache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l1licache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l2cache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0.l2cache.upstream[1]	PVBusSlave

## Ports for ARMCortexA520CT\_CortexA725CT\_CortexX925CT

Table 3-358: Ports

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[36]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[36]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[36]	Signal	Master	Timer signals to SOC
CNTHVIRQ[36]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[36]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[36]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[36]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[36]	Signal	Master	Timer signals to SOC.
commirq[36]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[36]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[36]	PChannel	Master	Core PCSM signals
core_powerdown_out[36]	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
coreerrirq[36]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[36]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[36]	Signal	Slave	Disable cryptography extensions after reset.
cti[36]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[36]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[36]	Signal	Master	No power-down request.
dbgpwrupreq[36]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.

Name	Protocol	Type	Description
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[36]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[36]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[36]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[36]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[36]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[36]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[36]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[36]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[36]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[36]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[36]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[36]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[36]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.

Name	Protocol	Type	Description
vfiq[36]	Signal	Slave	Virtualised FIQ.
virq[36]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[36]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM CortexA520CT\_CortexA725CT\_CortexX925CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

### AEND3\_DEFAULT

#### Type

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0



**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l3cache-size`****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**`l3cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l3cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l3cache-ways`****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**subcluster0.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster0.core\_cache\_protection****Type**

int

**Default value**

0x1

**Description**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0



**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**subcluster0.cpuX.l2cache-read\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x10

**Description**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`subcluster0.cpuX.l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`subcluster0.cpuX.l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`subcluster0.cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster0.cpuX.l2cache-ways`****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_bus\_width\_in\_bytes****Type**

int

**Default value**

0x20

**Description**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-prefetch\_enabled****Type**

bool



**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**subcluster0.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**subcluster0.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**subcluster0.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**subcluster0.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**subcluster0.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**subcluster0.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**subcluster0.ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster0.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster0.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster0.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster0.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster0.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster0.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster0.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster0.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster0.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster0.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster0.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect.  
1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster0.has\_actlr2****Type**

bool

**Default value**

0x1

**Description**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

**subcluster0.has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**subcluster0.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`subcluster0.icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster0.icache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**`subcluster0.invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**`subcluster0.memory_tagging_support_level`****Type**

int

**Default value**

0x3



**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster0.pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.ras\_extra\_configurations****Type**

string

**Default value**

```
"[ { \"Index\": 1, \"ERXMISCO_mask\": 0xFFFFc0003fc3, \"ERXMISC1_mask\": 0x03F870003FF30f07, \"ERXPFGCTL_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISCO_mask\": 0xFFFFe007ffc0, \"ERXMISCO_reset\": 0x2, \"ERXSTATUS_IERR_mask\": 0x300, \"ERXMISC1_mask\": 0x0FF8700FFFF31f0f, \"ERXPFGCTL_reset\": 0x1000 } ]"
```

**Description**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCn masks - these are 64 bit masks covering the 64 bit registers ERXMISCn\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

**subcluster0.ras\_pfg\_clock\_mhz****Type**

int

**Default value**

0xc

**Description**

RAS Pseudo-Fault generation clock rate in MHz.

**subcluster0.revision\_number****Type**

int

**Default value**

0x0

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster1.cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**subcluster1.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-size****Type**

int

**Default value**

0x100000

**Description**

L2 Cache size in bytes.

**subcluster1.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.l2cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`subcluster1.cpuX.max_code_cache_mb`****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.



**subcluster1.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.etc.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster1.ete.ETE\_REVISION****Type**

int

**Default value**

0x1

**Description**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

**subcluster1.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster1.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster1.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster1.ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster1.ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster1.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster1.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster1.ete.RETSTACK****Type**

int

**Default value**

0x1

**Description**

Return stack depth.

**subcluster1.ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster1.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster1.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster1.ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.



**subcluster1.ete.TSMARK****Type**

bool

**Default value**

0x1

**Description**

Whether timestamp markers are supported.

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster1.has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3). Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**subcluster1.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster1.has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**subcluster1.has\_statistical\_profiling****Type**

bool

**Default value**

0x0

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**subcluster1.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`subcluster1.icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`subcluster1.icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`subcluster1.instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**`subcluster1.invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster1.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster1.pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**subcluster1.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**subcluster1.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster1.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster2.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster2.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster2.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster2.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster2.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster2.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster2.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster2.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster2.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.



**subcluster2.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster2.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**subcluster2.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster2.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster2.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster2.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster2.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster2.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster2.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster2.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster2.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster2.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster2.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster2.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster2.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster2.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster2.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster2.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster2.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`subcluster2.dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit

or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

#### **`subcluster2.dcache-read_latency`**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

#### **`subcluster2.dcache-size`**

##### **Type**

int

##### **Default value**

0x10000

##### **Description**

L1 D-Cache size in bytes.

#### **`subcluster2.dcache-snoop_data_transfer_latency`**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

#### **`subcluster2.dcache-write_access_latency`**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be



used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

#### **subcluster2.dcache-write\_latency**

##### **Type**

int

##### **Default value**

0x0

##### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

#### **subcluster2.ecv\_support\_level**

##### **Type**

int

##### **Default value**

0x2

##### **Description**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

#### **subcluster2.ete.CLAIMTAGS**

##### **Type**

int

##### **Default value**

0x4

##### **Description**

Number of claim tags.

#### **subcluster2.ete.MAX\_INST\_PER\_Q**

##### **Type**

int

##### **Default value**

0x1

##### **Description**

Maximum limit for the number of instructions implied by a Q element.

#### **subcluster2.ete.NumberOfRSPairs**

##### **Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster2.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster2.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster2.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster2.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster2.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster2.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster2.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster2.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster2.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster2.ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x2

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**subcluster2.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster2.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster2.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster2.has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**subcluster2.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster2.has\_mt\_pmu\_disable\_feature****Type**

int

**Default value**

0x0

**Description**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

**subcluster2.has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**subcluster2.has\_v8\_7\_spe\_inverted\_filtering****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

**subcluster2.has\_v8\_7\_spe\_previous\_branch\_target****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

**subcluster2.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster2.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster2.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster2.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster2.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster2.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster2.icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**subcluster2.instruction\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**subcluster2.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster2.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster2.mpam\_has\_altsp****Type**

bool

**Default value**

0x0

**Description**

MPAM Whether MPAMIDR\_EL1.HAS\_ALTSP bit is set or clear.

**subcluster2.mpamidr\_has\_force\_ns****Type**

int

**Default value**

0x0



**Description**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**subcluster2.mpamidr\_has\_sdeflt****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**subcluster2.mpamidr\_has\_tidr****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**subcluster2.pmu-num\_counters****Type**

int

**Default value**

0x1f

**Description**

Number of PMU counters implemented.

**subcluster2.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**subcluster2.tcr\_txsz\_undersize\_should\_fault****Type**

bool

**Default value**

0x0

**Description**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

**subcluster2.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster2.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster2.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster2.walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.45 **ARMCortexA710CT**

ARMCortexA710CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-359: IP revisions support**

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA710CT**

The model supports the following features:

- DynamIQ Shared Unit-110 (DSU-110) system registers.
- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- PChannel for the cluster and for each core.
- BROADCASTPERSIST pin.
- Optional peripheral port.
- L3Cache partition.
- Per-core clock.
- Utility bus.
- AArch32 at EL0.

Support for the following features is planned for a future release:

- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (the Arm TLM2 extensions) bus protocols.
- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, and nPMBIRQ signals.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- Error correction or detection features.
- Self-test features (MBIST).
- Snoop filtering.

This model supports the Arm®v8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM Cortex-A710CT

This model has the following Iris instances:

**Table 3-360: ARM Cortex-A710CT Iris instances**

InstanceName	ComponentName
ARM Cortex-A710CT	Cluster_ARM_Cortex-A710
ARM Cortex-A710CT.AMU	PVBusLogger
ARM Cortex-A710CT.AMU.mapper	PVBusMapper
ARM Cortex-A710CT.DAP	PVBusLogger
ARM Cortex-A710CT.DAP.mapper	PVBusMapper
ARM Cortex-A710CT.DSU	DSU
ARM Cortex-A710CT.DSU.PPU_cluster	PPUv1
ARM Cortex-A710CT.DSU.PPU_cluster.busslave	PVBusSlave
ARM Cortex-A710CT.DSU.PPU_core0	PPUv1
ARM Cortex-A710CT.DSU.PPU_core0.busslave	PVBusSlave
ARM Cortex-A710CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex-A710CT.DSU.shared_cache	PVCache
ARM Cortex-A710CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM Cortex-A710CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM Cortex-A710CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM Cortex-A710CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARM Cortex-A710CT.DSU.utility_slave[0]	PVBusSlave
ARM Cortex-A710CT.MMAP	PVBusLogger

InstanceName	ComponentName
ARMCortexA710CT.MMAP.mapper	PVBusMapper
ARMCortexA710CT.RAS	PVBusLogger
ARMCortexA710CT.RAS.mapper	PVBusMapper
ARMCortexA710CT.cpu0	ARM_Cortex-A710
ARMCortexA710CT.cpu0.UTLB	TLB
ARMCortexA710CT.cpu0.debug_rom	debug_rom
ARMCortexA710CT.cpu0.dtlb	TLB
ARMCortexA710CT.cpu0.l1dcache	PVCache
ARMCortexA710CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA710CT.cpu0.l1icache	PVCache
ARMCortexA710CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA710CT.cpu0.l2cache	PVCache
ARMCortexA710CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA710CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA710CT.ext_bus	PVBusLogger
ARMCortexA710CT.ext_bus.mapper	PVBusMapper
ARMCortexA710CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA710CT.global_debug_rom	debug_rom
ARMCortexA710CT.secondary_debug_rom	debug_rom
ARMCortexA710CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-361: ARMCortexA710CT MTI instances**

InstanceName	ComponentName
ARMCortexA710CT.AMU	PVBusLogger
ARMCortexA710CT.AMU.mapper	PVBusMapper
ARMCortexA710CT.DAP	PVBusLogger
ARMCortexA710CT.DAP.mapper	PVBusMapper
ARMCortexA710CT.DSU	DSU
ARMCortexA710CT.DSU.PPU_cluster	PPUv1
ARMCortexA710CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA710CT.DSU.PPU_core0	PPUv1
ARMCortexA710CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA710CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA710CT.DSU.shared_cache	PVCache
ARMCortexA710CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA710CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA710CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA710CT.DSU.shared_cache.upstream[3]	PVBusSlave

InstanceName	ComponentName
ARMCortexA710CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA710CT.MMAP	PVBusLogger
ARMCortexA710CT.MMAP.mapper	PVBusMapper
ARMCortexA710CT.RAS	PVBusLogger
ARMCortexA710CT.RAS.mapper	PVBusMapper
ARMCortexA710CT.cpu0	ARM_Cortex-A710
ARMCortexA710CT.cpu0.UTLB	TLB
ARMCortexA710CT.cpu0.l1dcache	PVCache
ARMCortexA710CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA710CT.cpu0.l1icache	PVCache
ARMCortexA710CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA710CT.cpu0.l2cache	PVCache
ARMCortexA710CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA710CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA710CT.ext_bus	PVBusLogger
ARMCortexA710CT.ext_bus.mapper	PVBusMapper
ARMCortexA710CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA710CT

**Table 3-362: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.

Name	Protocol	Type	Description
cfgend[12]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[12]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[12]	Signal	Master	Timer signals to SOC
CNTHVIRQ[12]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[12]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[12]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[12]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[12]	Signal	Master	Timer signals to SOC.
commirq[12]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[12]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel[12]	PChannel	Master	Core PCSM signals
core_powerdown_out[12]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[12]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[12]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error

Name	Protocol	Type	Description
cryptodisable[12]	Signal	Slave	Disable cryptography extensions after reset.
cti[12]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[12]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[12]	Signal	Master	No power-down request.
dbgpwrupreq[12]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[12]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[12]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[12]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmuirq[12]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[12]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[12]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[12]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[12]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[12]	Signal	Slave	Per core System Error physical pins.



Name	Protocol	Type	Description
spiden	Signal	Slave	External debug interface.
ticks[12]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[12]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[12]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[12]	Signal	Slave	Virtualised FIQ.
virq[12]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[12]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A710CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0



**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

**etc.CLAIMTAGS****Type**

int

**Default value**

0x20

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x1

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.



**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x2

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**`force_zero_mpam_partid_and_pmg`****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**`has_acp`****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**`has_ete`****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**`has_peripheral_port`****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit

or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

### **l3cache-read\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

### **l3cache-size**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache size in bytes.

### **l3cache-snoop\_data\_transfer\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

### **l3cache-snoop\_issue\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.



**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x2

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.46 ARMCortexA715CT

ARMCortexA715CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-363: IP revisions support**

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About ARMCortexA715CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- DynamIQ r3p0.
- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.
- Utility bus.
- Support for Interrupt signals from SPE is added as pmbirq[8].

Support for the following features is planned for a future release:

- TRBE.
- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (TLM2 extensions from Arm) bus protocols.
- Core-Complex.
- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.



The `cfgsdisable` signal will be removed in a future release.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Limitations

- Some of the SVE instructions do not raise undefined exceptions when SP is used as a source register.

## Iris and MTI instances for ARMCortexA715CT

This model has the following Iris instances:

**Table 3-364: ARMCortexA715CT Iris instances**

InstanceName	ComponentName
ARMCortexA715CT	Cluster_ARM_Cortex-A715
ARMCortexA715CT.AMU	PVBusLogger
ARMCortexA715CT.AMU.mapper	PVBusMapper
ARMCortexA715CT.DAP	PVBusLogger
ARMCortexA715CT.DAP.mapper	PVBusMapper
ARMCortexA715CT.DSU	DSU
ARMCortexA715CT.DSU.PPU_cluster	PPUv1
ARMCortexA715CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA715CT.DSU.PPU_core0	PPUv1
ARMCortexA715CT.DSU.PPU_core0.busslave	PVBusSlave

InstanceName	ComponentName
ARMCortexA715CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA715CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA715CT.DSU.shared_cache	PVCache
ARMCortexA715CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA715CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA715CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA715CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA715CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA715CT.MMAP	PVBusLogger
ARMCortexA715CT.MMAP.mapper	PVBusMapper
ARMCortexA715CT.RAS	PVBusLogger
ARMCortexA715CT.RAS.mapper	PVBusMapper
ARMCortexA715CT.cpu0	ARM_Cortex-A715
ARMCortexA715CT.cpu0.UTLB	TLB
ARMCortexA715CT.cpu0.debug_rom	debug_rom
ARMCortexA715CT.cpu0.dtlb	TLB
ARMCortexA715CT.cpu0.l1dcache	PVCache
ARMCortexA715CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA715CT.cpu0.l1icache	PVCache
ARMCortexA715CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA715CT.cpu0.l2cache	PVCache
ARMCortexA715CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA715CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA715CT.ext_bus	PVBusLogger
ARMCortexA715CT.ext_bus.mapper	PVBusMapper
ARMCortexA715CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA715CT.global_debug_rom	debug_rom
ARMCortexA715CT.secondary_debug_rom	debug_rom
ARMCortexA715CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-365: ARMCortexA715CT MTI instances**

InstanceName	ComponentName
ARMCortexA715CT.AMU	PVBusLogger
ARMCortexA715CT.AMU.mapper	PVBusMapper
ARMCortexA715CT.DAP	PVBusLogger
ARMCortexA715CT.DAP.mapper	PVBusMapper
ARMCortexA715CT.DSU	DSU
ARMCortexA715CT.DSU.PPU_cluster	PPUv1

InstanceName	ComponentName
ARMCortexA715CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA715CT.DSU.PPU_core0	PPUv1
ARMCortexA715CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA715CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA715CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA715CT.DSU.shared_cache	PVCache
ARMCortexA715CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA715CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA715CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA715CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA715CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA715CT.MMAP	PVBusLogger
ARMCortexA715CT.MMAP.mapper	PVBusMapper
ARMCortexA715CT.RAS	PVBusLogger
ARMCortexA715CT.RAS.mapper	PVBusMapper
ARMCortexA715CT.cpu0	ARM_Cortex-A715
ARMCortexA715CT.cpu0.UTLB	TLB
ARMCortexA715CT.cpu0.l1dcache	PVCache
ARMCortexA715CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA715CT.cpu0.l1icache	PVCache
ARMCortexA715CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA715CT.cpu0.l2cache	PVCache
ARMCortexA715CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA715CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA715CT.ext_bus	PVBusLogger
ARMCortexA715CT.ext_bus.mapper	PVBusMapper
ARMCortexA715CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA715CT

**Table 3-366: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).

Name	Protocol	Type	Description
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[12]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[12]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[12]	Signal	Master	Timer signals to SOC
CNTHVIRQ[12]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[12]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[12]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[12]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[12]	Signal	Master	Timer signals to SOC.



Name	Protocol	Type	Description
commirq[12]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[12]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[12]	PChannel	Master	Core PCSM signals
core_powerdown_out[12]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[12]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[12]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[12]	Signal	Slave	Disable cryptography extensions after reset.
cti[12]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[12]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[12]	Signal	Master	No power-down request.
dbgprupreq[12]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[12]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[12]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[12]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[12]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[12]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[12]	Signal	Master	PPU core interrupt.

Name	Protocol	Type	Description
ppu_core_wakerequest[12]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[12]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[12]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[12]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[12]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[12]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[12]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[12]	Signal	Slave	Virtualised FIQ.
virq[12]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[12]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A715CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

**AEND1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

**AEND2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0



**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`cpuX.l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**  
A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC**

**Type**  
int

**Default value**  
0x123456

**Description**  
A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT**

**Type**  
int

**Default value**  
0x3c

**Description**  
T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC**

**Type**  
int

**Default value**  
0xab

**Description**  
T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line**

**Type**  
string

**Default value**  
""

**Description**  
Command line available to semihosting calls.

**cpuX.semihosting-cwd**

**Type**  
string

**Default value**  
""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int



**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`dcache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`default_opmode`****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**`diagnostics`****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**`enable_simulation_performance_optimizations`****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**error\_record\_feature\_register****Type**

string

**Default value**

```
"[{\"ED\":0x2,\"IMPDEF_3_2\":0x1,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x2,\"TS\":0x0,\"Visibility\":\"Cluster\"},{\"ED\":0x2,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0}]\"
```

**Description**

RAS feature register values. An array of JSON objects. The JSON schema for the array is: [{"ED":0x0, "IMPDEF\_3\_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other\_feature\_register\_values]. Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4. RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x1

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**`force_zero_mpam_partid_and_pmg`****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**`has_enhanced_pan`****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**`has_ete`****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**`has_large_va`****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA).  
Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x0

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.



**icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**instruction\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of  $n$  means the accumulator will use  $(n * \text{accumulator value})$  to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**pseudo\_fault\_generation\_feature\_register****Type**

string

**Default value**

```
"[{\"OF\":true, \"UC\":true, \"UEU\":false, \"UER\":false, \"UEO\":false, \"DE\":0x1, \"CE\":0x1,
  \"CI\":true, \"ER\":false, \"PN\":true, \"AV\":false, \"MV\":true, \"SYN\":true, \"R\":true},{\"OF
  \":false, \"UC\":true, \"UEU\":false, \"UER\":false, \"UEO\":false, \"DE\":0x1, \"CE\":0x1, \"CI
  \":false, \"ER\":false, \"PN\":false, \"AV\":false, \"MV\":false, \"SYN\":false, \"R\":true}]"
```

**Description**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_pseudo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.47 **ARMCortexA720AECT**

ARMCortexA720AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-367: IP revisions support**

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**AE-specific features implemented**

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.
- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.

As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, cpu0 and cpu1 identify the available cores and associated ports, not cpu0 and cpu2.

- Hybrid mode is not modeled in the DSU.

**Iris and MTI instances for ARMCortexA720AECT**

This model has the following Iris instances:

**Table 3-368: ARMCortexA720AECT Iris instances**

InstanceName	ComponentName
ARMCortexA720AECT	Cluster_ARM_Cortex-A720AE
ARMCortexA720AECT.AMU	PVBusLogger
ARMCortexA720AECT.AMU.mapper	PVBusMapper
ARMCortexA720AECT.DAP	PVBusLogger
ARMCortexA720AECT.DAP.mapper	PVBusMapper
ARMCortexA720AECT.DSU	DSU
ARMCortexA720AECT.DSU.PPU_cluster	PPUv1
ARMCortexA720AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA720AECT.DSU.PPU_core0	PPUv1
ARMCortexA720AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA720AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA720AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache	PVCache
ARMCortexA720AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA720AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA720AECT.MMAP	PVBusLogger
ARMCortexA720AECT.MMAP.mapper	PVBusMapper
ARMCortexA720AECT.RAS	PVBusLogger
ARMCortexA720AECT.RAS.mapper	PVBusMapper
ARMCortexA720AECT.cpu0	ARM_Cortex-A720AE
ARMCortexA720AECT.cpu0.UTLB	TLB
ARMCortexA720AECT.cpu0.debug_rom	debug_rom
ARMCortexA720AECT.cpu0.dtlb	TLB
ARMCortexA720AECT.cpu0.l1dcache	PVCache
ARMCortexA720AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA720AECT.cpu0.l1licache	PVCache
ARMCortexA720AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA720AECT.cpu0.l2cache	PVCache
ARMCortexA720AECT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA720AECT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA720AECT.ext_bus	PVBusLogger
ARMCortexA720AECT.ext_bus.mapper	PVBusMapper
ARMCortexA720AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA720AECT.global_debug_rom	debug_rom



InstanceName	ComponentName
ARMCortexA720AECT.secondary_debug_rom	debug_rom
ARMCortexA720AECT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-369: ARMCortexA720AECT MTI instances**

InstanceName	ComponentName
ARMCortexA720AECT.AMU	PVBusLogger
ARMCortexA720AECT.AMU.mapper	PVBusMapper
ARMCortexA720AECT.DAP	PVBusLogger
ARMCortexA720AECT.DAP.mapper	PVBusMapper
ARMCortexA720AECT.DSU	DSU
ARMCortexA720AECT.DSU.PPU_cluster	PPUv1
ARMCortexA720AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA720AECT.DSU.PPU_core0	PPUv1
ARMCortexA720AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA720AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA720AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache	PVCache
ARMCortexA720AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA720AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA720AECT.MMAP	PVBusLogger
ARMCortexA720AECT.MMAP.mapper	PVBusMapper
ARMCortexA720AECT.RAS	PVBusLogger
ARMCortexA720AECT.RAS.mapper	PVBusMapper
ARMCortexA720AECT.cpu0	ARM_Cortex-A720AE
ARMCortexA720AECT.cpu0.UTLB	TLB
ARMCortexA720AECT.cpu0.l1dcache	PVCache
ARMCortexA720AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA720AECT.cpu0.l1licache	PVCache
ARMCortexA720AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA720AECT.cpu0.l2cache	PVCache
ARMCortexA720AECT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA720AECT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA720AECT.ext_bus	PVBusLogger
ARMCortexA720AECT.ext_bus.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexA720AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA720AECT

**Table 3-370: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[14]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[14]	Signal	Master	Timer signals to SOC
CNTHVIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[14]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[14]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[14]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[14]	Signal	Master	Timer signals to SOC.
commirq[14]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[14]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[14]	PChannel	Master	Core PCSM signals
core_powerdown_out[14]	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
coreerrirq[14]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[14]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[14]	Signal	Slave	Disable cryptography extensions after reset.
cti[14]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[14]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[14]	Signal	Master	No power-down request.
dbgpwrupreq[14]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.

Name	Protocol	Type	Description
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[14]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[14]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[14]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[14]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[14]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[14]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[14]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[14]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[14]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[14]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[14]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[14]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[14]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.

Name	Protocol	Type	Description
vfiq[14]	Signal	Slave	Virtualised FIQ.
virq[14]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[14]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA720AECT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

### AEND3\_DEFAULT

#### Type

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1



**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**cluster\_split\_lock\_config****Type**

int

**Default value**

0x1

**Description**

Default SPLIT/LOCKED config. The valid values are: 1 - Only LOCKED mode support, 4 - Only SPLIT mode support, 5 - SPLIT or MIXED mode support. Valid only when enable\_ae\_features is true.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**`cpuX.semihosting-T32_HLT`****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**`cpuX.semihosting-Thumb_SVC`****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**`cpuX.semihosting-cmd_line`****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**`cpuX.semihosting-cwd`****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**`cpuX.semihosting-enable`****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**`cpuX.semihosting-heap_base`****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**`cpuX.semihosting-heap_limit`****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**`cpuX.semihosting-stack_base`****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**`cpuX.semihosting-stack_limit`****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**`cpuX.trace_special_hlt_imm16`****Type**

int

**Default value**

0xf000



**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_lock\_step****Type**

bool

**Default value**

0x0

**Description**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase

differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.ETE\_REVISION****Type**

int

**Default value**

0x1

**Description**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CMOD****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CMOD value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**ete.TSMARK****Type**

bool

**Default value**

0x1

**Description**

Whether timestamp markers are supported.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.



**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect.  
1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_impdef\_transient\_fault\_protection****Type**

bool

**Default value**

0x1

**Description**

Support the Transient Fault Protection (TFP) flop parity errors through RAS registers (FEAT\_TFP).

**has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x0

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.



**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**`treat_PAC_as_NOP`**

**Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**`walk_cache_latency`**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.48 ARMCortexA720CT**

ARMCortexA720CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-371: IP revisions support**

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA720CT**

The model supports the following features:

- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- Internal PPU support is present.

- A P-Channel for the cluster and for each core.
- `BROADCASTPERSIST` pin.
- Optional peripheral port.
- Memory-mapped register access to MPAM.
- Per-core clock.
- Utility bus.

Support for the following features is planned for a future release:

- DSU-120 system features are not fully implemented.
- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (TLM2 extensions from Arm) bus protocols.
- Core-Complex.
- `BROADCASTCACHEMAINTPOU` pin
- `COREINSTRRET` and `COREINSTRRUN` signals.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not supported but signals are implemented.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARMCortexA720CT

This model has the following Iris instances:

**Table 3-372: ARMCortexA720CT Iris instances**

InstanceName	ComponentName
ARMCortexA720CT	Cluster_ARM_Cortex-A720

InstanceName	ComponentName
ARMCortexA720CT.AMU	PVBusLogger
ARMCortexA720CT.AMU.mapper	PVBusMapper
ARMCortexA720CT.DAP	PVBusLogger
ARMCortexA720CT.DAP.mapper	PVBusMapper
ARMCortexA720CT.DSU	DSU
ARMCortexA720CT.DSU.PPU_cluster	PPUv1
ARMCortexA720CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA720CT.DSU.PPU_core0	PPUv1
ARMCortexA720CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA720CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA720CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA720CT.DSU.shared_cache	PVCache
ARMCortexA720CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA720CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA720CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA720CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA720CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA720CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA720CT.MMAP	PVBusLogger
ARMCortexA720CT.MMAP.mapper	PVBusMapper
ARMCortexA720CT.RAS	PVBusLogger
ARMCortexA720CT.RAS.mapper	PVBusMapper
ARMCortexA720CT.cpu0	ARM_Cortex-A720
ARMCortexA720CT.cpu0.UTLB	TLB
ARMCortexA720CT.cpu0.debug_rom	debug_rom
ARMCortexA720CT.cpu0.dtlb	TLB
ARMCortexA720CT.cpu0.l1dcache	PVCache
ARMCortexA720CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA720CT.cpu0.l1icache	PVCache
ARMCortexA720CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA720CT.cpu0.l2cache	PVCache
ARMCortexA720CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA720CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA720CT.ext_bus	PVBusLogger
ARMCortexA720CT.ext_bus.mapper	PVBusMapper
ARMCortexA720CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA720CT.global_debug_rom	debug_rom
ARMCortexA720CT.secondary_debug_rom	debug_rom
ARMCortexA720CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-373: ARMCortexA720CT MTI instances**

InstanceName	ComponentName
ARMCortexA720CT.AMU	PVBusLogger
ARMCortexA720CT.AMU.mapper	PVBusMapper
ARMCortexA720CT.DAP	PVBusLogger
ARMCortexA720CT.DAP.mapper	PVBusMapper
ARMCortexA720CT.DSU	DSU
ARMCortexA720CT.DSU.PPU_cluster	PPUv1
ARMCortexA720CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA720CT.DSU.PPU_core0	PPUv1
ARMCortexA720CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA720CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA720CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA720CT.DSU.shared_cache	PVCache
ARMCortexA720CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA720CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA720CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA720CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA720CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA720CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA720CT.MMAP	PVBusLogger
ARMCortexA720CT.MMAP.mapper	PVBusMapper
ARMCortexA720CT.RAS	PVBusLogger
ARMCortexA720CT.RAS.mapper	PVBusMapper
ARMCortexA720CT.cpu0	ARM_Cortex-A720
ARMCortexA720CT.cpu0.UTLB	TLB
ARMCortexA720CT.cpu0.l1dcache	PVCache
ARMCortexA720CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA720CT.cpu0.l1icache	PVCache
ARMCortexA720CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA720CT.cpu0.l2cache	PVCache
ARMCortexA720CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA720CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA720CT.ext_bus	PVBusLogger
ARMCortexA720CT.ext_bus.mapper	PVBusMapper
ARMCortexA720CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA720CT

Table 3-374: Ports

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[14]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[14]	Signal	Master	Timer signals to SOC
CNTHVIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[14]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[14]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[14]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[14]	Signal	Master	Timer signals to SOC.
commirq[14]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[14]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[14]	PChannel	Master	Core PCSM signals
core_powerdown_out[14]	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
coreerrirq[14]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[14]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[14]	Signal	Slave	Disable cryptography extensions after reset.
cti[14]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[14]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[14]	Signal	Master	No power-down request.
dbgpwrupreq[14]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.

Name	Protocol	Type	Description
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[14]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[14]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[14]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[14]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[14]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[14]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[14]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[14]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[14]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[14]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[14]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[14]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[14]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.



Name	Protocol	Type	Description
vfiq[14]	Signal	Slave	Virtualised FIQ.
virq[14]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[14]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexA720CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

### AEND3\_DEFAULT

#### Type

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CFGTE`****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**`cpuX.CRYPTODISABLE`****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**`cpuX.RVBARADDR`****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0



**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.ETE\_REVISION****Type**

int



**Default value**

0x1

**Description**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**ete.TSMARK****Type**

bool

**Default value**

0x1

**Description**

Whether timestamp markers are supported.

**`force_mte_tag_access_razwi_and_ignore_tag_checks`****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**`force_zero_PSTATE_PAN`****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**`force_zero_mpam_partid_and_pmg`****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**`has_enhanced_pan`****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3). Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x0

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**`l3cache-has_mpam`****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**`l3cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**`l3cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.



**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_acp****Type**

int

**Default value**

0x0

**Description**

Number of ACP ports.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**`treat_PAC_as_NOP`**

**Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**`walk_cache_latency`**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.49 ARMCortexA725CT**

ARMCortexA725CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-375: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexA725CT**

A DSU-120 DynamIQ cluster containing a configurable number of Cortex-A725 cores.

The number of cores in the cluster is configurable using the following parameter:

**NUM\_CORES**

Possible values are 1-14

The following DSU/CPU features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

**Iris and MTI instances for ARM Cortex-A725CT**

This model has the following Iris instances:

**Table 3-376: ARM Cortex-A725CT Iris instances**

InstanceName	ComponentName
ARM Cortex-A725CT	Cluster_ARM_Cortex-A725
ARM Cortex-A725CT.AMU	PVBusLogger
ARM Cortex-A725CT.AMU.mapper	PVBusMapper
ARM Cortex-A725CT.DAP	PVBusLogger
ARM Cortex-A725CT.DAP.mapper	PVBusMapper
ARM Cortex-A725CT.DSU	DSU
ARM Cortex-A725CT.DSU.PPU_cluster	PPUv1
ARM Cortex-A725CT.DSU.PPU_cluster.busslave	PVBusSlave
ARM Cortex-A725CT.DSU.PPU_core0	PPUv1
ARM Cortex-A725CT.DSU.PPU_core0.busslave	PVBusSlave
ARM Cortex-A725CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM Cortex-A725CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex-A725CT.DSU.shared_cache	PVCache
ARM Cortex-A725CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARM Cortex-A725CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARM Cortex-A725CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARM Cortex-A725CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARM Cortex-A725CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARM Cortex-A725CT.DSU.utility_slave[0]	PVBusSlave
ARM Cortex-A725CT.MMAP	PVBusLogger

InstanceName	ComponentName
ARMCortexA725CT.MMAP.mapper	PVBusMapper
ARMCortexA725CT.RAS	PVBusLogger
ARMCortexA725CT.RAS.mapper	PVBusMapper
ARMCortexA725CT.cpu0	ARM_Cortex-A725
ARMCortexA725CT.cpu0.UTLB	TLB
ARMCortexA725CT.cpu0.debug_rom	debug_rom
ARMCortexA725CT.cpu0.dtlb	TLB
ARMCortexA725CT.cpu0.l1dcache	PVCache
ARMCortexA725CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA725CT.cpu0.l1icache	PVCache
ARMCortexA725CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA725CT.cpu0.l2cache	PVCache
ARMCortexA725CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA725CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA725CT.ext_bus	PVBusLogger
ARMCortexA725CT.ext_bus.mapper	PVBusMapper
ARMCortexA725CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA725CT.global_debug_rom	debug_rom
ARMCortexA725CT.secondary_debug_rom	debug_rom
ARMCortexA725CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-377: ARMCortexA725CT MTI instances**

InstanceName	ComponentName
ARMCortexA725CT.AMU	PVBusLogger
ARMCortexA725CT.AMU.mapper	PVBusMapper
ARMCortexA725CT.DAP	PVBusLogger
ARMCortexA725CT.DAP.mapper	PVBusMapper
ARMCortexA725CT.DSU	DSU
ARMCortexA725CT.DSU.PPU_cluster	PPUv1
ARMCortexA725CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA725CT.DSU.PPU_core0	PPUv1
ARMCortexA725CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA725CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA725CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA725CT.DSU.shared_cache	PVCache
ARMCortexA725CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA725CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA725CT.DSU.shared_cache.upstream[2]	PVBusSlave

InstanceName	ComponentName
ARMCortexA725CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA725CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA725CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA725CT.MMAP	PVBusLogger
ARMCortexA725CT.MMAP.mapper	PVBusMapper
ARMCortexA725CT.RAS	PVBusLogger
ARMCortexA725CT.RAS.mapper	PVBusMapper
ARMCortexA725CT.cpu0	ARM_Cortex-A725
ARMCortexA725CT.cpu0.UTLB	TLB
ARMCortexA725CT.cpu0.l1dcache	PVCache
ARMCortexA725CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA725CT.cpu0.l1icache	PVCache
ARMCortexA725CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA725CT.cpu0.l2cache	PVCache
ARMCortexA725CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA725CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA725CT.ext_bus	PVBusLogger
ARMCortexA725CT.ext_bus.mapper	PVBusMapper
ARMCortexA725CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

## Ports for ARMCortexA725CT

**Table 3-378: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.



Name	Protocol	Type	Description
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[14]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[14]	Signal	Master	Timer signals to SOC
CNTHVIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[14]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[14]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[14]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[14]	Signal	Master	Timer signals to SOC.
commirq[14]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[14]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel[14]	PChannel	Master	Core PCSM signals
core_powerdown_out[14]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info

Name	Protocol	Type	Description
coreerrirq[14]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[14]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[14]	Signal	Slave	Disable cryptography extensions after reset.
cti[14]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[14]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[14]	Signal	Master	No power-down request.
dbgpwrupreq[14]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[14]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[14]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[14]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[14]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[14]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[14]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[14]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[14]	Signal	Slave	Per core RAM Error Interrupt.

Name	Protocol	Type	Description
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[14]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[14]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[14]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[14]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[14]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[14]	Signal	Slave	Virtualised FIQ.
virq[14]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[14]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A725CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

**AEND2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CFGTE`****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**`cpuX.CRYPTODISABLE`****Type**

bool



**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x100000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0



**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**`dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

### **dcache-write\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

### **default\_opmode**

#### **Type**

int

#### **Default value**

0x4

#### **Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

### **diagnostics**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Enable DynamIQ diagnostic messages.

### **enable\_simulation\_performance\_optimizations**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.ETE\_REVISION****Type**

int

**Default value**

0x1

**Description**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x1

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**ete.TSMARK****Type**

bool

**Default value**

0x1

**Description**

Whether timestamp markers are supported.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect.  
1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x0

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.



**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**instruction\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l3cache-size`****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**`l3cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l3cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l3cache-ways`****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases

where execution of quantums in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**`treat_PAC_as_NOP`**

**Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**`walk_cache_latency`**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.50 ARMCortexA725CT\_CortexX925CT

ARMCortexA725CT\_CortexX925CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-379: IP revisions support

Revision	Quality level
CortexA725 r0p0	Preliminary support
CortexX925 r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexA720CT\_CortexX925CT

The number of cores in each subcluster is configurable using the following parameters:

**`subcluster0.NUM_CORES`**

Possible values are 1-13 (ARMCortexA720CT).

**`subcluster1.NUM_CORES`**

Possible values are 1-13 (ARMCortexX925CT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- `<port_name>[0-12]` for cores in `subcluster0`.
- `<port_name>[13-25]` for cores in `subcluster1`.



All instances in the Master cross trigger matrix port array, `cti[26]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu12` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu12` identify cores in `subcluster1`.

For information about the cores in this model, see:

- [ARMCortexA720CT](#).
- [ARMCortexX925CT](#).

## Iris and MTI instances for ARMCortexA725CT\_CortexX925CT

This model has the following Iris instances:

**Table 3-380: ARMCortexA725CT\_CortexX925CT Iris instances**

InstanceName	ComponentName
ARMCortexA725CT_CortexX925CT	Cluster_ARM_CortexA725_CortexX925_Heterogeneous
ARMCortexA725CT_CortexX925CT.AMU	PVBusLogger
ARMCortexA725CT_CortexX925CT.AMU.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.DAP	PVBusLogger
ARMCortexA725CT_CortexX925CT.DAP.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.DSU	DSU
ARMCortexA725CT_CortexX925CT.DSU.PPU_cluster	PPUv1
ARMCortexA725CT_CortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.PPU_core0	PPUv1
ARMCortexA725CT_CortexX925CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.PPU_core1	PPUv1
ARMCortexA725CT_CortexX925CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA725CT_CortexX925CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache	PVCache
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[0]	PVBusSlave



InstanceName	ComponentName
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.MMAP	PVBusLogger
ARMCortexA725CT_CortexX925CT.MMAP.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.RAS	PVBusLogger
ARMCortexA725CT_CortexX925CT.RAS.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.cpu0.debug_rom	debug_rom
ARMCortexA725CT_CortexX925CT.cpu1.debug_rom	debug_rom
ARMCortexA725CT_CortexX925CT.ext_bus	PVBusLogger
ARMCortexA725CT_CortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA725CT_CortexX925CT.global_debug_rom	debug_rom
ARMCortexA725CT_CortexX925CT.secondary_debug_rom	debug_rom
ARMCortexA725CT_CortexX925CT.subcluster0	Subcluster_ARM_Cortex-A725
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0	ARM_Cortex-A725
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.dtlb	TLB
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster0.sve	ScalableVectorExtension
ARMCortexA725CT_CortexX925CT.subcluster1	Subcluster_ARM_Cortex-X925
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0	ARM_Cortex-X925
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l1licache	PVCache

InstanceName	ComponentName
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster1.cpu1.dtlb	TLB
ARMCortexA725CT_CortexX925CT.subcluster1.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-381: ARMCortexA725CT\_CortexX925CT MTI instances**

InstanceName	ComponentName
ARMCortexA725CT_CortexX925CT.AMU	PVBusLogger
ARMCortexA725CT_CortexX925CT.AMU.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.DAP	PVBusLogger
ARMCortexA725CT_CortexX925CT.DAP.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.DSU	DSU
ARMCortexA725CT_CortexX925CT.DSU.PPU_cluster	PPUv1
ARMCortexA725CT_CortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.PPU_core0	PPUv1
ARMCortexA725CT_CortexX925CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.PPU_core1	PPUv1
ARMCortexA725CT_CortexX925CT.DSU.PPU_core1.busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA725CT_CortexX925CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache	PVCache
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[5]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[6]	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.MMAP	PVBusLogger
ARMCortexA725CT_CortexX925CT.MMAP.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.RAS	PVBusLogger
ARMCortexA725CT_CortexX925CT.RAS.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.ext_bus	PVBusLogger

InstanceName	ComponentName
ARMCortexA725CT_CortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0	ARM_Cortex-A725
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.UTLB	TLB
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1licache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0	ARM_Cortex-X925
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.UTLB	TLB
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l1dcache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l1licache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache	PVCache
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0.l2cache.upstream[1]	PVBusSlave

## Ports for ARMCortexA725CT\_CortexX925CT

**Table 3-382: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.

Name	Protocol	Type	Description
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[26]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[26]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[26]	Signal	Master	Timer signals to SOC
CNTHVIRQ[26]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[26]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[26]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[26]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[26]	Signal	Master	Timer signals to SOC.
commirq[26]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[26]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel[26]	PChannel	Master	Core PCSM signals
core_powerdown_out[26]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info

Name	Protocol	Type	Description
coreerrirq[26]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[26]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[26]	Signal	Slave	Disable cryptography extensions after reset.
cti[26]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[26]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[26]	Signal	Master	No power-down request.
dbgpwrupreq[26]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[26]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[26]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[26]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[26]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[26]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[26]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[26]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[26]	Signal	Slave	Per core RAM Error Interrupt.

Name	Protocol	Type	Description
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[26]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[26]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[26]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[26]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[26]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[26]	Signal	Slave	Virtualised FIQ.
virq[26]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[26]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A725CT\_Cortex X925CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

**AEND2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0



**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate

the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantums in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**subcluster0.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster0.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster0.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster0.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster0.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster0.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster0.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster0.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster0.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster0.cpuX.force-fpsid****Type**

bool



**Default value**

0x1

**Description**

Override the FPSID value.

**subcluster0.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-

read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

### **subcluster0.cpuX.l2cache-read\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

### **subcluster0.cpuX.l2cache-size**

#### **Type**

int

#### **Default value**

0x100000

#### **Description**

L2 Cache size in bytes.

### **subcluster0.cpuX.l2cache-snoop\_data\_transfer\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

### **subcluster0.cpuX.l2cache-snoop\_issue\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster0.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster0.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster0.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster0.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster0.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster0.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster0.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster0.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**subcluster0.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster0.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster0.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster0.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster0.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster0.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster0.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster0.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**`subcluster0.dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-write_access_latency`****Type**

int



**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`subcluster0.dcache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`subcluster0.ete.CLAIMTAGS`****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**`subcluster0.ete.ETE_REVISION`****Type**

int

**Default value**

0x1

**Description**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

**`subcluster0.ete.MAX_INST_PER_Q`****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster0.etc.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster0.etc.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster0.etc.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**subcluster0.etc.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**subcluster0.etc.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster0.etc.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster0.etc.RETSTACK****Type**

int

**Default value**

0x1

**Description**

Return stack depth.

**subcluster0.etc.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**subcluster0.etc.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster0.etc.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster0.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster0.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster0.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster0.ete.TSMARK****Type**

bool

**Default value**

0x1

**Description**

Whether timestamp markers are supported.

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster0.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster0.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster0.has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**subcluster0.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster0.has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**subcluster0.has\_statistical\_profiling****Type**

bool

**Default value**

0x0

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**subcluster0.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster0.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster0.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster0.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster0.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster0.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster0.icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**subcluster0.instruction\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**subcluster0.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster0.pmu-num\_counters****Type**

int

**Default value**

0x6



**Description**

Number of PMU counters implemented.

**subcluster0.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**subcluster0.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster0.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster0.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster0.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**subcluster1.NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

**subcluster1.cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**subcluster1.cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**subcluster1.cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**subcluster1.cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**subcluster1.cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**subcluster1.cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**subcluster1.cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**subcluster1.cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**subcluster1.cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**subcluster1.cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**subcluster1.cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**subcluster1.cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**subcluster1.cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**subcluster1.cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**subcluster1.cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**subcluster1.cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**subcluster1.cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.



**subcluster1.cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**subcluster1.cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**subcluster1.cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**subcluster1.cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**subcluster1.cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**subcluster1.cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**subcluster1.dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**subcluster1.dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**subcluster1.dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**subcluster1.dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**subcluster1.ecv\_support\_level****Type**

int

**Default value**

0x2

**Description**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

**subcluster1.ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**subcluster1.ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**subcluster1.ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**subcluster1.ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**subcluster1.ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**subcluster1.ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**subcluster1.ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**subcluster1.ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**subcluster1.ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**subcluster1.ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**subcluster1.ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x2

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**subcluster1.force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**subcluster1.has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**subcluster1.has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**subcluster1.has\_mt\_pmu\_disable\_feature****Type**

int

**Default value**

0x0



**Description**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

**subcluster1.has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**subcluster1.has\_v8\_7\_spe\_inverted\_filtering****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

**subcluster1.has\_v8\_7\_spe\_previous\_branch\_target****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

**subcluster1.icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**subcluster1.icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**subcluster1.icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**subcluster1.icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**subcluster1.icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**subcluster1.icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**subcluster1.instruction\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**subcluster1.memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**subcluster1.mpam\_has\_altsp****Type**

bool

**Default value**

0x0

**Description**

MPAM Whether MPAMIDR\_EL1.HAS\_ALTSP bit is set or clear.

**subcluster1.mpamidr\_has\_force\_ns****Type**

int

**Default value**

0x0

**Description**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**subcluster1.mpamidr\_has\_sdeflt****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_SDEFILT bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**subcluster1.mpamidr\_has\_tidr****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear Possible values of this parameter are:  
- 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**subcluster1.pmu-num\_counters****Type**

int

**Default value**

0x1f

**Description**

Number of PMU counters implemented.

**subcluster1.ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**subcluster1.tcr\_txsz\_undersize\_should\_fault****Type**

bool

**Default value**

0x0

**Description**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

**subcluster1.tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**subcluster1.tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**subcluster1.treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**subcluster1.walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.51 ARMCortexM0CT

ARMCortexM0CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-383: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Differences between the model and the RTL

This model does not have a parameter that is equivalent to the RAR integration option. The architecturally-required register state is reset.

This model exposes a VTOR register through CADI but this register does not exist in the IP.

Armv6-M is a subset of Armv7-M. Arm does not guarantee that all Armv7-M-specific behavior is absent from Armv6-M Fast Models cores. Therefore, Arm does not guarantee that code that runs on Armv7-M cores but fails on Armv6-M cores will also fail on Armv6-M Fast Models cores.

#### Iris and MTI instances for ARMCortexM0CT

This model has the following Iris instances:

**Table 3-384: ARMCortexM0CT Iris instances**

InstanceName	ComponentName
ARMCortexM0CT	ARM_Cortex-M0
ARMCortexM0CT.acp_mapper	PVBusMapper
ARMCortexM0CT.ext_bus	PVBusLogger
ARMCortexM0CT.ext_bus.mapper	PVBusMapper
ARMCortexM0CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-385: ARMCortexM0CT MTI instances**

InstanceName	ComponentName
ARMCortexM0CT	ARM_Cortex-M0
ARMCortexM0CT.acp_mapper	PVBusMapper
ARMCortexM0CT.ext_bus	PVBusLogger
ARMCortexM0CT.ext_bus.mapper	PVBusMapper
ARMCortexM0CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM0CT

Table 3-386: Ports

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
currpri	Value	Master	Current execution priority.
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr[32]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
lockup	Signal	Master	Asserted when the processor is in lockup state.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
stcalib	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARMCortexM0CT

**BIGENDINIT****Type**

bool

**Default value**

0x0

**Description**

Initialize processor to big endian mode.

**BKPT****Type**

int

**Default value**

0x4



**Description**

Number of breakpoint unit comparators implemented.

**DBG****Type**

bool

**Default value**

0x1

**Description**

Set whether debug extensions are implemented.

**NUM\_IRQ****Type**

int

**Default value**

0x20

**Description**

Number of user interrupts.

**SYST****Type**

bool

**Default value**

0x1

**Description**

Enable support for SysTick timer functionality.

**WIC****Type**

bool

**Default value**

0x1

**Description**

Include support for WIC-mode deep sleep.

**WPT****Type**

int

**Default value**

0x2

**Description**

Number of watchpoint unit comparators implemented.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**`master_id`****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**`min_sync_level`****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**`reported_patch_level`****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base****Type**

int

**Default value**

0x20800000

**Description**

Virtual address of base of descending stack.

**semihosting-stack\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of stack limit.

### 3.5.52 ARMCortexM0PlusCT

ARMCortexM0PlusCT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-387: IP revisions support**

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Differences between the model and the RTL**

This model does not have a parameter that is equivalent to the RAR integration option. The architecturally required register state is reset.

This model does not have a parameter that is equivalent to the RAR integration option. The architecturally-required register state is reset.

This model exposes a VTOR register through CADI but this register does not exist in the IP.

Armv6-M is a subset of Armv7-M. Arm does not guarantee that all Armv7-M-specific behavior is absent from Armv6-M Fast Models cores. Therefore, Arm does not guarantee that code that runs on Armv7-M cores but fails on Armv6-M cores will also fail on Armv6-M Fast Models cores.

**Iris and MTI instances for ARMCortexM0PlusCT**

This model has the following Iris instances:

**Table 3-388: ARMCortexM0PlusCT Iris instances**

InstanceName	ComponentName
ARMCortexM0PlusCT	ARM_Cortex-M0+
ARMCortexM0PlusCT.acp_mapper	PVBusMapper
ARMCortexM0PlusCT.ext_bus	PVBusLogger
ARMCortexM0PlusCT.ext_bus.mapper	PVBusMapper
ARMCortexM0PlusCT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-389: ARMCortexM0PlusCT MTI instances**

InstanceName	ComponentName
ARMCortexM0PlusCT	ARM_Cortex-M0+

InstanceName	ComponentName
ARMCortexM0PlusCT.acp_mapper	PVBusMapper
ARMCortexM0PlusCT.ext_bus	PVBusLogger
ARMCortexM0PlusCT.ext_bus.mapper	PVBusMapper
ARMCortexM0PlusCT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM0PlusCT

**Table 3-390: Ports**

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cpuwait	Signal	Slave	CPUWAIT extends effect of reset when true
currpri	Value	Master	Current execution priority.
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr[32]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
io_port_in	PVBus	Slave	I/O port pair. See the documentation for the io_port_out port.
io_port_out	PVBus	Master	I/O port pair. Used if IOP is true. Transactions from io_port_out which do not "match" should be returned via io_port_in. For performance reasons, the I/O port interface is not modelled directly. Instead, a simple PVBus gasket is inserted at the point in the memory system where the I/O port would be. In hardware, a device would be attached to the port which would tell the CPU whether it would like to intercept each transaction, given its address. This can be modelled in a performant manner by connecting a PVBusMapper-based device to io_port_out which intercepts transactions of interest and passes other transactions back to the CPU via io_port_in. Your I/O port device model is also responsible for aborting transactions which would be aborted on hardware (e.g. exclusives) if necessary.
lockup	Signal	Master	Asserted when the processor is in lockup state.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
stcalib	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARMCortexM0PlusCT

### **BIGENDINIT**

**Type**

bool

**Default value**

0x0

**Description**

Initialize processor to big endian mode.

### **BKPT**

**Type**

int

**Default value**

0x4

**Description**

Number of breakpoint unit comparators implemented.

### **DBG**

**Type**

bool

**Default value**

0x1

**Description**

Set whether debug extensions are implemented.

### **IOP**

**Type**

bool

**Default value**

0x0

**Description**

Send all d-side transactions to the port, io\_port\_out. Transactions which do not match should be returned to the port, io\_port\_in.

### **IRQDIS**

**Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n].

**NUM\_IRQ****Type**

int

**Default value**

0x20

**Description**

Number of user interrupts.

**NUM\_MPU\_REGION****Type**

int

**Default value**

0x0

**Description**

Number of MPU regions.

**SYST****Type**

bool

**Default value**

0x1

**Description**

Enable support for SysTick timer functionality.

**USER****Type**

bool

**Default value**

0x0

**Description**

Enable support for Unprivileged/Privileged Extension.

**VTOR****Type**

bool

**Default value**

0x0



**Description**

Include Vector Table Offset Register.

**WIC****Type**

bool

**Default value**

0x1

**Description**

Include support for WIC-mode deep sleep.

**WPT****Type**

int

**Default value**

0x2

**Description**

Number of watchpoint unit comparators implemented.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**`master_id`****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base**

**Type**

int

**Default value**

0x20800000

**Description**

Virtual address of base of descending stack.

**semihosting-stack\_limit**

**Type**

int

**Default value**

0x20700000

**Description**

Virtual address of stack limit.

**3.5.53 ARM CortexM3CT**

ARM CortexM3CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-391: IP revisions support**

Revision	Quality level
r2p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Differences between the model and the RTL**

This component has the following differences from the corresponding revision of the RTL implementation:

- The WIC is not currently implemented.
- Power control is not implemented, so the processor does not set the SLEEPING or SLEEPDEEP signals. It does not support powering down of the processor.
- Only the minimal level of debug support is provided (no DAP, FPB, DWT, or halting debug capability).
- Debug-related components are not implemented.

- The unimplemented registers are the processor debug registers, system debug registers, debug interface port registers, TPIU registers, and ETM registers.
- The processor must still be clocked even if it has asserted the sleeping or sleepdeep signals.
- Disabling processor features using the Auxiliary Control Register is not supported.
- Only a single pvbus\_m master port is provided. This combines the ICode, DCode, and System bus interfaces of the RTL. The external PPB bus is provided by the pv\_ppbus\_m master port.
- In privileged mode, STRT and LDRT to the PPB region are not forbidden access.
- No support for ETM, TPIU, or HTM.
- There is no supported equivalent of the RESET\_ALL\_REGS configuration setting in RTL (that forces all registers to have a well-defined value on reset).
- The RTL implements the ROM table as an external component on the External Private Peripheral Bus. In the CT model, the ROM table is implemented internally as a fallback if an external PPB access in the ROM table address region aborts. This permits the default ROM table to be overridden (by implementing an external component connected to the external PPB to handle accesses to these addresses) without requiring every user of the processor to implement and connect a ROM table component.

### Implementation of ITM in M-class models

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an `ITM_PACKET_TYPE` field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

### Iris and MTI instances for ARMCortexM3CT

This model has the following Iris instances:

**Table 3-393: ARMCortexM3CT Iris instances**

InstanceName	ComponentName
ARMCortexM3CT	ARM_Cortex-M3
ARMCortexM3CT.acp_mapper	PVBusMapper
ARMCortexM3CT.ext_bus	PVBusLogger
ARMCortexM3CT.ext_bus.mapper	PVBusMapper
ARMCortexM3CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-394: ARMCortexM3CT MTI instances**

InstanceName	ComponentName
ARMCortexM3CT	ARM_Cortex-M3
ARMCortexM3CT.acp_mapper	PVBusMapper
ARMCortexM3CT.ext_bus	PVBusLogger
ARMCortexM3CT.ext_bus.mapper	PVBusMapper
ARMCortexM3CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM3CT

**Table 3-395: Ports**

Name	Protocol	Type	Description
ahb_ap	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
currpri	Value	Master	Current execution priority.
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr[240]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
lockup	Signal	Master	Asserted when the processor is in lockup state.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
stcalib	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).

Name	Protocol	Type	Description
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARMCortexM3CT

### BB\_PRESENT

#### Type

bool

#### Default value

0x1

#### Description

Enable bitbanding.

### BIGENDINIT

#### Type

bool

#### Default value

0x0

#### Description

Initialize processor to big endian mode.

### DBG\_LVL

#### Type

int

#### Default value

0x3

#### Description

0: No debug; 1: Minimal debug; 2: Full debug without DWT address matching; 3: Full debug support with DWT data-comparators.

### LVL\_WIDTH

#### Type

int

#### Default value

0x3

#### Description

Number of bits of interrupt priority.

**NUM\_IRQ****Type**

int

**Default value**

0x10

**Description**

Number of user interrupts.

**NUM\_MPU\_REGION****Type**

int

**Default value**

0x8

**Description**

Number of MPU regions.

**TRACE\_LVL****Type**

int

**Default value**

0x1

**Description**

Level of instrumentation trace supported. 0: No trace, 1: Standard trace. ITM, TPIU and DWT triggers and counters present, 2: Full trace. ITM, TPIU, ETM and DWT triggers and counters present.

**WIC****Type**

bool

**Default value**

0x1

**Description**

Include support for WIC-mode deep sleep.

**cpi\_div****Type**

int

**Default value**

0x1



**Description**

divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**master\_id****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**`0xffffffffffffffff`**Description**

Purely cosmetic revision number value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**`semihosting-Thumb_SVC`****Type**`int`**Default value**`0xab`**Description**

T32 SVC number for semihosting.

**`semihosting-cmd_line`****Type**`string`**Default value**`""`**Description**

Command line available to semihosting SVC calls.

**`semihosting-cwd`****Type**`string`**Default value**`""`**Description**

Base directory for semihosting file access.

**`semihosting-enable`****Type**`bool`**Default value**`0x1`**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base****Type**

int

**Default value**

0x20800000

**Description**

Virtual address of base of descending stack.

**semihosting-stack\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of stack limit.

### 3.5.54 ARMCortexM4CT

ARMCortexM4CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-396: IP revisions support**

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- The Wakeup Interrupt Controller (WIC) is not implemented.
- Power control is not implemented. Powering down of the processor is not supported. The processor must still be clocked even if it has asserted the sleeping or sleepdeep signals.
- Only the minimal level of debug support is provided (no DAP, FPB, DWT, or halting debug capability).
- No debug-related components are implemented.
- The unimplemented registers are the processor debug registers, system debug registers, debug interface port registers, TPIU registers, and ETM registers.
- No support for ETM, TPIU, or HTM.
- There is no supported equivalent of the RESET\_ALL\_REGS configuration setting in RTL (that forces all registers to have a well-defined value on reset).
- Disabling processor features using the Auxiliary Control Register is not supported.
- Only a single pbus\_m master port is provided. This combines the ICode, DCode, and System bus interfaces of the RTL. The external PPB bus is provided by the pv\_ppbus\_m master port.
- In privileged mode, STRT and LDRT to the PPB region are not forbidden access.
- The RTL implements the ROM table as an external component on the External Private Peripheral Bus. In the CT model, the ROM table is implemented internally as a fallback if an external PPB access in the ROM table address region aborts. This permits the default ROM table to be overridden (by implementing an external component connected to the external PPB to handle accesses to these addresses) without requiring every user of the processor to implement and connect a ROM table component.
- Because the CT model does not provide a DAP port or halting debug capability, the dbgen signal is ignored.

### Implementation of ITM in M-class models

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an `ITM_PACKET_TYPE` field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.

Field value	Description	Supported by model
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

## Iris and MTI instances for ARMCortexM4CT

This model has the following Iris instances:

**Table 3-398: ARMCortexM4CT Iris instances**

InstanceName	ComponentName
ARMCortexM4CT	ARM_Cortex-M4
ARMCortexM4CT.acp_mapper	PVBusMapper
ARMCortexM4CT.ext_bus	PVBusLogger
ARMCortexM4CT.ext_bus.mapper	PVBusMapper
ARMCortexM4CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-399: ARMCortexM4CT MTI instances**

InstanceName	ComponentName
ARMCortexM4CT	ARM_Cortex-M4
ARMCortexM4CT.acp_mapper	PVBusMapper
ARMCortexM4CT.ext_bus	PVBusLogger
ARMCortexM4CT.ext_bus.mapper	PVBusMapper
ARMCortexM4CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM4CT

**Table 3-400: Ports**

Name	Protocol	Type	Description
ahb_ap	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.

Name	Protocol	Type	Description
currpri	Value	Master	Current execution priority.
dbgen	Signal	Slave	Disallow (DAP) debugger access.
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpudisable	Signal	Slave	Configure core with no FPU on reset.
fpxxc	Value	Master	Port that sends the value of the FPxxC exception flags (FPIX, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
intisr[240]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
lockup	Signal	Master	Asserted when the processor is in lockup state.
mpudisable	Signal	Slave	Configure core with no MPU on reset.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
stcalib	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARM CortexM4CT

### BB\_PRESENT

#### Type

bool

#### Default value

0x1

#### Description

Enable bitbanding.

### BIGENDINIT

#### Type

bool

#### Default value

0x0

**Description**

Initialize processor to big endian mode.

**DBGLVL****Type**

int

**Default value**

0x3

**Description**

0: No debug; 1: Minimal debug; 2: Full debug without DWT address matching; 3: Full debug support with, DWT can compare data as well as address.

**LVL\_WIDTH****Type**

int

**Default value**

0x3

**Description**

Number of bits of interrupt priority.

**NUM\_IRQ****Type**

int

**Default value**

0x10

**Description**

Number of user interrupts.

**NUM\_MPU\_REGION****Type**

int

**Default value**

0x8

**Description**

Number of MPU regions.

**TRACE\_LVL****Type**

int

**Default value**

0x1

**Description**

Level of instrumentation trace supported. 0: No trace, 1: Standard trace. ITM, TPIU and DWT triggers and counters present, 2: Full trace. ITM, TPIU, ETM and DWT triggers and counters present.

**WIC****Type**

bool

**Default value**

0x1

**Description**

Include support for WIC-mode deep sleep.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**`master_id`****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.



**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**`semihosting-cwd`****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**`semihosting-enable`****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**`semihosting-heap_base`****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**`semihosting-heap_limit`****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base**

**Type**  
int

**Default value**  
0x20800000

**Description**  
Virtual address of base of descending stack.

**semihosting-stack\_limit**

**Type**  
int

**Default value**  
0x20700000

**Description**  
Virtual address of stack limit.

**vfp-present**

**Type**  
bool

**Default value**  
0x1

**Description**  
Set whether the model has VFP support.

3.5.55 ARMCortexM7CT

ARMCortexM7CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-401: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Differences between the model and the RTL

- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- ECC support is hardware-specific so is not modeled.

## Implementation of ITM in M-class models

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an `ITM_PACKET_TYPE` field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

## Iris and MTI instances for ARM CortexM7CT

This model has the following Iris instances:

**Table 3-403: ARM CortexM7CT Iris instances**

InstanceName	ComponentName
ARM CortexM7CT	ARM_Cortex-M7
ARM CortexM7CT.acp_mapper	PVBusMapper
ARM CortexM7CT.ext_bus	PVBusLogger
ARM CortexM7CT.ext_bus.mapper	PVBusMapper
ARM CortexM7CT.l1_incoherent_interconnect	PVCache
ARM CortexM7CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARM CortexM7CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARM CortexM7CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARM CortexM7CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARM CortexM7CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARM CortexM7CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARM CortexM7CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARM CortexM7CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARM CortexM7CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARM CortexM7CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARM CortexM7CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave

InstanceName	ComponentName
ARMCortexM7CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexM7CT.l1dcache	PVCache
ARMCortexM7CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM7CT.l1icache	PVCache
ARMCortexM7CT.l1icache.upstream[0]	PVBusSlave
ARMCortexM7CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-404: ARMCortexM7CT MTI instances**

InstanceName	ComponentName
ARMCortexM7CT	ARM_Cortex-M7
ARMCortexM7CT.acp_mapper	PVBusMapper
ARMCortexM7CT.ext_bus	PVBusLogger
ARMCortexM7CT.ext_bus.mapper	PVBusMapper
ARMCortexM7CT.l1_incoherent_interconnect	PVCache
ARMCortexM7CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexM7CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave

InstanceName	ComponentName
ARMCortexM7CT.l1dcache	PVCache
ARMCortexM7CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM7CT.l1icache	PVCache
ARMCortexM7CT.l1icache.upstream[0]	PVBusSlave
ARMCortexM7CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM7CT

**Table 3-405: Ports**

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
ahbp_m	PVBus	Master	The core will generate Vendor System data accesses on this port.
ahbs	PVBus	Slave	External master (e.g. DMA) can write TCMs (whether or not enabled in xTCMCR).
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cpuwait	Signal	Slave	When this signal is HIGH out of reset, it forces the processor into a quiescent state that delays its boot-up sequence and instruction execution until this signal is driven LOW
currpri	Value	Master	Current execution priority.
dap_s	PVBus	Slave	Debug Access Port (DAP).
dbgen	Signal	Slave	Invasive debug enable.
dbgrestart	Signal	Slave	External debug request.
dbgrestarted	Signal	Master	External debug request.
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpudisable	Signal	Slave	Configure core with no FPU on reset.
fpxxc	Value	Master	Port that sends the value of the FPxxC exception flags (FPIX, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	Signal	Master	External debug request.
initahbpen	Signal	Slave	Enable AHBP on the next reset
initvtor	Value	Slave	Initial value of the Vector Table Offset Register (VTOR)
intisr[240]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
lockup	Signal	Master	Asserted when the processor is in lockup state.
mpudisable	Signal	Slave	Configure core with no MPU on reset.
niden	Signal	Slave	Non-invasive debug enable.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	PVBus	Master	The core will generate bus requests on this port.

Name	Protocol	Type	Description
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
stcalib	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARM Cortex M7CT

### **BIGENDINIT**

#### Type

bool

#### Default value

0x0

#### Description

Initialize processor to big endian mode.

### **CFG\_AHBPSZ**

#### Type

int

#### Default value

0x0

#### Description

Size of the AHB port memory region. 0=AHBP disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

### **CTI**

#### Type

bool

#### Default value

0x0

#### Description

CTI (Cross Trigger Interface) included.

### **CTI\_irq0\_pin**

#### Type

int

**Default value**

0x4

**Description**

CTI interrupt request 0 pin.

**CTI\_irq1\_pin****Type**

int

**Default value**

0x5

**Description**

CTI interrupt request 1 pin.

**DBGLVL****Type**

int

**Default value**

0x1

**Description**

0: 2 DWT, 4 FPB; 1: 4 DWT, 8 FPB comparators.

**DP\_FLOAT****Type**

bool

**Default value**

0x1

**Description**

Support 8-byte floats.

**INITAHBPEN****Type**

bool

**Default value**

0x0

**Description**

The AHBP enable state at reset.

**INITVTOR****Type**

int



**Default value**

0x0

**Description**

vector-table offset at reset.

**LVL\_WIDTH****Type**

int

**Default value**

0x3

**Description**

Number of bits of interrupt priority.

**NUM\_IRQ****Type**

int

**Default value**

0x20

**Description**

Number of user interrupts.

**NUM\_MPU\_REGION****Type**

int

**Default value**

0x10

**Description**

Number of MPU regions.

**TRC****Type**

bool

**Default value**

0x1

**Description**

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

**WIC****Type**

bool

**Default value**

0x1

**Description**

Include support for WIC-mode deep sleep.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**`dcache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 D-cache size in bytes.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dtcm\_enable****Type**

bool

**Default value**

0x0

**Description**

Enable DTCM at reset.

**dtcm\_size****Type**

int

**Default value**

0x100

**Description**

DTCM size in KB.

**icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**itcm\_enable****Type**

bool

**Default value**

0x0

**Description**

Enable ITCM at reset.

**itcm\_size****Type**

int

**Default value**

0x100

**Description**

ITCM size in KB.

**master\_id****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base****Type**

int

**Default value**

0x20800000

**Description**

Virtual address of base of descending stack.

**semihosting-stack\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of stack limit.

**vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

### 3.5.56 ARMCortexM23CT

ARMCortexM23CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-406: IP revisions support**

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Differences between the CT model and RTL implementations

The model does not support MTB, ETM, or TPIU. MTB RAM is absent on the model.

#### Iris and MTI instances for ARMCortexM23CT

This model has the following Iris instances:

**Table 3-407: ARMCortexM23CT Iris instances**

InstanceName	ComponentName
ARMCortexM23CT	ARM_Cortex-M23
ARMCortexM23CT.acp_mapper	PVBusMapper
ARMCortexM23CT.ext_bus	PVBusLogger
ARMCortexM23CT.ext_bus.mapper	PVBusMapper
ARMCortexM23CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-408: ARMCortexM23CT MTI instances**

InstanceName	ComponentName
ARMCortexM23CT	ARM_Cortex-M23
ARMCortexM23CT.acp_mapper	PVBusMapper
ARMCortexM23CT.ext_bus	PVBusLogger
ARMCortexM23CT.ext_bus.mapper	PVBusMapper
ARMCortexM23CT.l2_flusher	AsyncCacheFlushUnit

#### Ports for ARMCortexM23CT

**Table 3-409: Ports**

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.

Name	Protocol	Type	Description
core_dside_bus_gasket_in	PVBus	Slave	-
core_dside_bus_gasket_out	PVBus	Master	-
cpuwait	Signal	Slave	Clear = Core goes through reset sequence as normal, Set = Core waits out of reset.
currpri	Value	Master	Current execution priority.
dap_s	PVBus	Slave	Debug Access Port (DAP).
dbgen	Signal	Slave	Invasive debug control signals. Debug enable, Set=enabled, Clear=disabled
dbgrestart	Signal	Slave	External request to leave debug state
dbgrestarted	Signal	Master	Acknowledge for DBGRESTART
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
halted	Signal	Master	Core is in halt mode debug state
hreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
idau	PVBus	Master	The core will generate IDAU requests on this port.
idau_invalidate_region	Value_64	Slave	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
initvtor	Value	Slave	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initvtorns	Value	Slave	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
io_port_in	PVBus	Slave	I/O port pair. See the documentation for the io_port_out port.
io_port_out	PVBus	Master	I/O port pair. Used if IOP is true. Transactions from io_port_out which do not "match" should be returned via io_port_in. For performance reasons, the I/O port interface is not modelled directly. Instead, a simple PVBus gasket is inserted at the point in the memory system where the I/O port would be. In hardware, a device would be attached to the port which would tell the CPU whether it would like to intercept each transaction, given its address. This can be modelled in a performant manner by connecting a PVBusMapper-based device to io_port_out which intercepts transactions of interest and passes other transactions back to the CPU via io_port_in. Your I/O port device model is also responsible for aborting transactions which would be aborted on hardware (e.g. exclusives) if necessary.
irq[240]	Signal	Slave	This signal array delivers signals to the NVIC.
lockup	Signal	Master	Asserted when the processor is in lockup state.
niden	Signal	Slave	Non-invasive debug enable, Set=enabled, Clear=disabled
nmi	Signal	Slave	Configure non maskable interrupt.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.



Name	Protocol	Type	Description
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
spiden	Signal	Slave	Secure Debug enable , Set=enabled, Clear=disabled
spniden	Signal	Slave	Secure Non-invasive debug enable, Set=enabled, Clear=disabled
stcalib	Value	Slave	This is the calibration value for the Secure (or only, when ARMv8-M Security Extensions are not included) SysTick timer.
stcalibns	Value	Slave	This is the calibration value for the Non-Secure SysTick timer. When ARMv8-M Security Extensions are not included, this port will be ignored.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	Signal	Master	Acknowledge signal for WICENREQ
wicenreq	Signal	Slave	Request for deep sleep to be WIC-based deep sleep.
wicsense[483]	Signal	Master	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARMCortexM23CT

### BE

#### Type

bool

#### Default value

0x0

#### Description

Initialize processor to big endian mode.

### BKPT

#### Type

int

#### Default value

0x4

#### Description

Number of breakpoint unit comparators implemented.

### CTI

#### Type

bool

**Default value**

0x0

**Description**

CTI (Cross Trigger Interface) included.

**CTI\_irq0\_pin****Type**

int

**Default value**

0x4

**Description**

CTI interrupt request 0 pin.

**CTI\_irq1\_pin****Type**

int

**Default value**

0x5

**Description**

CTI interrupt request 1 pin.

**DBG****Type**

bool

**Default value**

0x1

**Description**

Set whether debug extensions are implemented.

**INITVTOR****Type**

int

**Default value**

0x0

**Description**

Secure vector-table offset at reset.

**INITVTORNS****Type**

int

**Default value**

0x0

**Description**

Non-Secure vector-table offset at reset.

**IOP****Type**

bool

**Default value**

0x0

**Description**

Send all d-side transactions to the port, io\_port\_out. Transactions which do not match should be returned to the port, io\_port\_in.

**IRQDIS0****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

**IRQDIS1****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

**IRQDIS2****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

**IRQDIS3****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

**IRQDIS4****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

**IRQDIS5****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

**IRQDIS6****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

**IRQDIS7****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

**MPU\_NS****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

**MPU\_S****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

**NUMIRQ****Type**

int

**Default value**

0x10

**Description**

Number of user interrupts.

**SAU****Type**

int

**Default value**

0x4

**Description**

Number of SAU regions (0 => no SAU).

**SAU\_CTRL.ALLNS****Type**

bool

**Default value**

0x0

**Description**

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

**SAU\_CTRL.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU at reset.

**SAU\_REGION0.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region0 at reset.

**SAU\_REGION0.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region0 at reset.

**SAU\_REGION0.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region0 at reset.

**SAU\_REGION0.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region0 at reset.

**SAU\_REGION1.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region1 at reset.

**SAU\_REGION1.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region1 at reset.

**SAU\_REGION1.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region1 at reset.

**SAU\_REGION1.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region1 at reset.

**SAU\_REGION2.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region2 at reset.

**SAU\_REGION2.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region2 at reset.

**SAU\_REGION2.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region2 at reset.

**SAU\_REGION2.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region2 at reset.

**SAU\_REGION3.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region3 at reset.

**SAU\_REGION3.ENABLE****Type**

bool

**Default value**

0x0



**Description**

Enable SAU region3 at reset.

**SAU\_REGION3.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region3 at reset.

**SAU\_REGION3.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region3 at reset.

**SAU\_REGION4.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region4 at reset.

**SAU\_REGION4.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region4 at reset.

**SAU\_REGION4.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region4 at reset.

**SAU\_REGION4.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region4 at reset.

**SAU\_REGION5.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region5 at reset.

**SAU\_REGION5.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region5 at reset.

**SAU\_REGION5.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region5 at reset.

**SAU\_REGION5.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region5 at reset.

**SAU\_REGION6.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region6 at reset.

**SAU\_REGION6.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region6 at reset.

**SAU\_REGION6.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region6 at reset.

**SAU\_REGION6.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region6 at reset.

**SAU\_REGION7.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region7 at reset.

**SAU\_REGION7.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region7 at reset.

**SAU\_REGION7.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region7 at reset.

**SAU\_REGION7.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region7 at reset.

**SECEXT****Type**

bool

**Default value**

0x1

**Description**

Whether the ARMv8-M Security Extensions are included.

**SYST****Type**

int

**Default value**

0x2

**Description**

Include SysTick timer functionality (0=Absent, 1=Secure only, 2=Secure and NS).

**VTOR****Type**

bool

**Default value**

0x1

**Description**

Include Vector Table Offset Register.

**WIC****Type**

bool

**Default value**

0x1

**Description**

Include support for WIC-mode deep sleep.

**WICLINES****Type**

int

**Default value**

0x12

**Description**

Number of lines supported by the WIC interface.

**WPT****Type**

int

**Default value**

0x4

**Description**

Number of watchpoint unit comparators implemented.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`**

**Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**`has_core_dside_bus_gasket`**

**Type**

bool

**Default value**

0x0

**Description**

STL gasket enabled.

**`master_id`**

**Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**`min_sync_level`**

**Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**`reported_patch_level`**

**Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base****Type**

int

**Default value**

0x20800000

**Description**

Virtual address of base of descending stack.

**semihosting-stack\_limit****Type**

int

**Default value**

0x20700000



**Description**

Virtual address of stack limit.

**3.5.57 ARMCortexM33CT**

ARMCortexM33CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-410: IP revisions support**

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Differences between the model and the RTL**

The model does not support the following:

- ETM, MTB, CTI, or TPIU. MTB RAM is absent on the model.
- The power control (Q-Channel) interface.

**Implementation of ITM in M-class models**

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an `ITM_PACKET_TYPE` field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

**Iris and MTI instances for ARMCortexM33CT**

This model has the following Iris instances:

**Table 3-412: ARMCortexM33CT Iris instances**

InstanceName	ComponentName
ARMCortexM33CT	ARM_Cortex-M33
ARMCortexM33CT.acp_mapper	PVBusMapper
ARMCortexM33CT.ext_bus	PVBusLogger
ARMCortexM33CT.ext_bus.mapper	PVBusMapper
ARMCortexM33CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-413: ARMCortexM33CT MTI instances**

InstanceName	ComponentName
ARMCortexM33CT	ARM_Cortex-M33
ARMCortexM33CT.acp_mapper	PVBusMapper
ARMCortexM33CT.ext_bus	PVBusLogger
ARMCortexM33CT.ext_bus.mapper	PVBusMapper
ARMCortexM33CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM33CT

**Table 3-414: Ports**

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	CoprocBusProtocol	Slave	Co-Processor Interface
cpuwait	Signal	Slave	Stall the CPU out of reset
currns	Signal	Master	Current Security state of the processor
currpri	Value	Master	Current execution priority.
dbgen	Signal	Slave	Invasive debug enable
dbgrestart	Signal	Slave	Request for synchronised exit from halt mode
dbgrestarted	Signal	Master	Handshakes with DBGRESTART
edbgrq	Signal	Slave	External request to enter halt mode
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	Value	Master	Port that sends the value of the FPxxC exception flags (FPIXC, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	Signal	Master	Indicates that the processor is in halt mode
idau	PVBus	Master	The core will generate IDAU requests on this port.

Name	Protocol	Type	Description
idau_invalidate_region	Value_64	Slave	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
initnsvtor	Value	Slave	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initsvtor	Value	Slave	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
intnum	Value	Master	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
irq[480]	Signal	Slave	This signal array delivers signals to the NVIC.
locknsmpu	Signal	Slave	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	Signal	Slave	Cortex-M33-specific LOCKNSVTOR, LOCKSVTAIRCR, LOCKSMPU, LOCKNSMPU, LOCKSAU. Disable writes to VTOR_NS
locksau	Signal	Slave	Disable writes to the SAU_* registers
locksmpu	Signal	Slave	Disable writes to the Secure MPU_* registers
locksvtaircr	Signal	Slave	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
lockup	Signal	Master	Asserted when the processor is in lockup state.
niden	Signal	Slave	Non-invasive debug enable
nmi	Signal	Slave	Configure non maskable interrupt.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
spiden	Signal	Slave	Secure invasive debug enable
spniden	Signal	Slave	Secure non-invasive debug enable
stcalib[2]	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	Signal	Master	Acknowledge signal for WICENREQ
wicenreq	Signal	Slave	Request for deep sleep to be WIC-based deep sleep.
wicsense[483]	Signal	Master	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARMCortexM33CT

### **BIGENDINIT**

**Type**

bool

**Default value**

0x0

**Description**

Initialize processor to big endian mode.

### **CDEMAPPEDONCP**

**Type**

int

**Default value**

0xff

**Description**

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

### **CDERTLID**

**Type**

int

**Default value**

0x20

**Description**

Value of ID\_AFR0.CDERTLID.

### **CFGNOCDECP**

**Type**

int

**Default value**

0x0

**Description**

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

### **CPIF**

**Type**

bool

**Default value**

0x1

**Description**

Specifies whether the external coprocessor interface is included.

**CPNSPRESENT****Type**

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

**CPSPPRESENT****Type**

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

**CTI****Type**

bool

**Default value**

0x0

**Description**

CTI (Cross Trigger Interface) included.

**CTI\_irq0\_pin****Type**

int

**Default value**

0x4

**Description**

CTI interrupt request 0 pin.

**CTI\_irq1\_pin****Type**

int

**Default value**

0x5

**Description**

CTI interrupt request 1 pin.

**DBGLVL****Type**

int

**Default value**

0x2

**Description**

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators.

**DSP****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has the DSP extension.

**FPU****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**INITNSVTOR****Type**

int

**Default value**

0x0

**Description**

Non-Secure vector-table offset at reset.

**INITSVTOR****Type**

int

**Default value**

0x0

**Description**

Secure vector-table offset at reset.

**IRQDIS0****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

**IRQDIS1****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

**IRQDIS10****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

**IRQDIS11****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

**IRQDIS12****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

**IRQDIS13****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

**IRQDIS14****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

**IRQDIS2****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

**IRQDIS3****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

**IRQDIS4****Type**

int



**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

**IRQDIS5****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

**IRQDIS6****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

**IRQDIS7****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

**IRQDIS8****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

**IRQDIS9****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

**IRQLVL****Type**

int

**Default value**

0x3

**Description**

Number of bits of interrupt priority.

**ITM****Type**

bool

**Default value**

0x1

**Description**

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

**LOCK\_NS\_MPU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Non-Secure MPU registers write.

**LOCK\_SAU****Type**

bool

**Default value**

0x0

**Description**

Lock down of SAU registers write.

**LOCK\_S\_MPU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Secure MPU registers write.

**MPU\_NS****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

**MPU\_S****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

**NUMIRQ****Type**

int

**Default value**

0x20

**Description**

Number of user interrupts.

**SAU****Type**

int

**Default value**

0x4

**Description**

Number of SAU regions (0 => no SAU).

**SAU\_CTRL.ALLNS****Type**

bool

**Default value**

0x0

**Description**

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

**SAU\_CTRL.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU at reset.

**SAU\_REGION0.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region0 at reset.

**SAU\_REGION0.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region0 at reset.

**SAU\_REGION0.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region0 at reset.

**SAU\_REGION0.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region0 at reset.

**SAU\_REGION1.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region1 at reset.

**SAU\_REGION1.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region1 at reset.

**SAU\_REGION1.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region1 at reset.

**SAU\_REGION1.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region1 at reset.

**SAU\_REGION2.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region2 at reset.

**SAU\_REGION2.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region2 at reset.

**SAU\_REGION2.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region2 at reset.

**SAU\_REGION2.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region2 at reset.

**SAU\_REGION3.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region3 at reset.

**SAU\_REGION3.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region3 at reset.

**SAU\_REGION3.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region3 at reset.

**SAU\_REGION3.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region3 at reset.

**SAU\_REGION4.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region4 at reset.

**SAU\_REGION4.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region4 at reset.

**SAU\_REGION4.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region4 at reset.

**SAU\_REGION4.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region4 at reset.

**SAU\_REGION5.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region5 at reset.

**SAU\_REGION5.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region5 at reset.

**SAU\_REGION5.LADDR****Type**

int

**Default value**

0x0



**Description**

Limit address of SAU region5 at reset.

**SAU\_REGION5.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region5 at reset.

**SAU\_REGION6.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region6 at reset.

**SAU\_REGION6.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region6 at reset.

**SAU\_REGION6.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region6 at reset.

**SAU\_REGION6.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region6 at reset.

**SAU\_REGION7.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region7 at reset.

**SAU\_REGION7.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region7 at reset.

**SAU\_REGION7.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region7 at reset.

**SAU\_REGION7.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region7 at reset.

**SECEXT****Type**

bool

**Default value**

0x1

**Description**

Whether the ARMv8-M Security Extensions are included.

**WIC****Type**

bool

**Default value**

0x1

**Description**

Include support for WIC-mode deep sleep.

**WICLINES****Type**

int

**Default value**

0x23

**Description**

Number of lines supported by the WIC interface.

**cde\_impl\_name****Type**

string

**Default value**

""

**Description**

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**has\_cde****Type**

bool

**Default value**

0x0

**Description**

Enables Custom Datapath Extensions.

**master\_id****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**`0xffffffffffffffff`**Description**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**`semihosting-Thumb_SVC`****Type**`int`**Default value**`0xab`**Description**

T32 SVC number for semihosting.

**`semihosting-cmd_line`****Type**`string`**Default value**`""`**Description**

Command line available to semihosting SVC calls.

**`semihosting-cwd`****Type**`string`**Default value**`""`**Description**

Base directory for semihosting file access.

**`semihosting-enable`****Type**`bool`**Default value**`0x1`**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base****Type**

int

**Default value**

0x20800000

**Description**

Virtual address of base of descending stack.

**semihosting-stack\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of stack limit.

**vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**3.5.58 ARMCortexM35PCT**

ARMCortexM35PCT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-415: IP revisions support**

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Differences between the model and the RTL**

- The model does not support the following:
  - ETM, MTB, CTI, or TPIU. MTB RAM is absent on the model.
  - Caches.
  - The co-processor interface.
  - The power control (Q-Channel) interface.
- The model does not implement any physical security features.
- Bits[3:0] of the Anti-tampering Features Control Register are supported for read/write. No functionality is implemented.
- Read/write access to the Anti-tampering Features Control Register is supported using SECKEY. No functionality is implemented.

**Implementation of ITM in M-class models**

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an `ITM_PACKET_TYPE` field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.

Field value	Description	Supported by model
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

## Iris and MTI instances for ARMCortexM35PCT

This model has the following Iris instances:

**Table 3-417: ARMCortexM35PCT Iris instances**

InstanceName	ComponentName
ARMCortexM35PCT	ARM_Cortex-M35P
ARMCortexM35PCT.acp_mapper	PVBusMapper
ARMCortexM35PCT.ext_bus	PVBusLogger
ARMCortexM35PCT.ext_bus.mapper	PVBusMapper
ARMCortexM35PCT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-418: ARMCortexM35PCT MTI instances**

InstanceName	ComponentName
ARMCortexM35PCT	ARM_Cortex-M35P
ARMCortexM35PCT.acp_mapper	PVBusMapper
ARMCortexM35PCT.ext_bus	PVBusLogger
ARMCortexM35PCT.ext_bus.mapper	PVBusMapper
ARMCortexM35PCT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM35PCT

**Table 3-419: Ports**

Name	Protocol	Type	Description
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	CoprocBusProtocol	Slave	Co-Processor Interface
cpuwait	Signal	Slave	Stall the CPU out of reset
currns	Signal	Master	Current Security state of the processor
currpri	Value	Master	Current execution priority.
dbgen	Signal	Slave	Invasive debug enable



Name	Protocol	Type	Description
dbgrestart	Signal	Slave	Request for synchronised exit from halt mode
dbgrestorted	Signal	Master	Handshakes with DBGRESTART
edbgrq	Signal	Slave	External request to enter halt mode
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	Value	Master	Port that sends the value of the FPxxC exception flags (FPIXC, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	Signal	Master	Indicates that the processor is in halt mode
idau	PVBus	Master	The core will generate IDAU requests on this port.
idau_invalidate_region	Value_64	Slave	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
initnsvtor	Value	Slave	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initsvtor	Value	Slave	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
intnum	Value	Master	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
irq[480]	Signal	Slave	This signal array delivers signals to the NVIC.
LOCKATFCR	Signal	Slave	Port Lock ATFCR register
locknsmpu	Signal	Slave	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	Signal	Slave	Disable writes to VTOR_NS
locksau	Signal	Slave	Disable writes to the SAU_* registers
locksmpu	Signal	Slave	Disable writes to the Secure MPU_* registers
locksvtaircr	Signal	Slave	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
lockup	Signal	Master	Asserted when the processor is in lockup state.
niden	Signal	Slave	Non-invasive debug enable
nmi	Signal	Slave	Configure non maskable interrupt.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
spiden	Signal	Slave	Secure invasive debug enable
spniden	Signal	Slave	Secure non-invasive debug enable
stcalib[2]	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.

Name	Protocol	Type	Description
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	Signal	Master	Acknowledge signal for WICENREQ
wicenreq	Signal	Slave	Request for deep sleep to be WIC-based deep sleep.
wicsense[483]	Signal	Master	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARM CortexM35PCT

### ATFINITEN

#### Type

bool

#### Default value

0x0

#### Description

ATFCR is enabled when the core goes out of reset.

### BIGENDINIT

#### Type

bool

#### Default value

0x0

#### Description

Initialize processor to big endian mode.

### CPIF

#### Type

bool

#### Default value

0x1

#### Description

Specifies whether the external coprocessor interface is included.

### CPNSPRESENT

#### Type

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

**CPSPRESENT****Type**

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

**CTI****Type**

bool

**Default value**

0x0

**Description**

CTI (Cross Trigger Interface) included.

**CTI\_irq0\_pin****Type**

int

**Default value**

0x4

**Description**

CTI interrupt request 0 pin.

**CTI\_irq1\_pin****Type**

int

**Default value**

0x5

**Description**

CTI interrupt request 1 pin.

**DBGLVL****Type**

int

**Default value**

0x2

**Description**

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators.

**DSP****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has the DSP extension.

**FPU****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**INITNSVTOR****Type**

int

**Default value**

0x0

**Description**

Non-Secure vector-table offset at reset.

**INITSVTOR****Type**

int

**Default value**

0x0

**Description**

Secure vector-table offset at reset.

**IRQDIS0****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

**IRQDIS1****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

**IRQDIS10****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

**IRQDIS11****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

**IRQDIS12****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

**IRQDIS13****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

**IRQDIS14****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

**IRQDIS2****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

**IRQDIS3****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

**IRQDIS4****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

**IRQDIS5****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

**IRQDIS6****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

**IRQDIS7****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

**IRQDIS8****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

**IRQDIS9****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

**IRQLVL****Type**

int

**Default value**

0x3

**Description**

Number of bits of interrupt priority.

**ITM****Type**

bool

**Default value**

0x1

**Description**

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

**LOCK\_NS\_MPU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Non-Secure MPU registers write.

**LOCK\_SAU****Type**

bool

**Default value**

0x0

**Description**

Lock down of SAU registers write.

**LOCK\_S\_MPU****Type**

bool

**Default value**

0x0



**Description**

Lock down of Secure MPU registers write.

**MPU\_NS****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

**MPU\_S****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

**NUMIRQ****Type**

int

**Default value**

0x20

**Description**

Number of user interrupts.

**SAU****Type**

int

**Default value**

0x4

**Description**

Number of SAU regions (0 => no SAU).

**SAU\_CTRL.ALLNS****Type**

bool

**Default value**

0x0

**Description**

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

**SAU\_CTRL.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU at reset.

**SAU\_REGION0.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region0 at reset.

**SAU\_REGION0.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region0 at reset.

**SAU\_REGION0.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region0 at reset.

**SAU\_REGION0.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region0 at reset.

**SAU\_REGION1.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region1 at reset.

**SAU\_REGION1.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region1 at reset.

**SAU\_REGION1.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region1 at reset.

**SAU\_REGION1.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region1 at reset.

**SAU\_REGION2.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region2 at reset.

**SAU\_REGION2.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region2 at reset.

**SAU\_REGION2.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region2 at reset.

**SAU\_REGION2.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region2 at reset.

**SAU\_REGION3.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region3 at reset.

**SAU\_REGION3.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region3 at reset.

**SAU\_REGION3.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region3 at reset.

**SAU\_REGION3.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region3 at reset.

**SAU\_REGION4.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region4 at reset.

**SAU\_REGION4.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region4 at reset.

**SAU\_REGION4.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region4 at reset.

**SAU\_REGION4.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region4 at reset.

**SAU\_REGION5.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region5 at reset.

**SAU\_REGION5.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region5 at reset.

**SAU\_REGION5.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region5 at reset.

**SAU\_REGION5.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region5 at reset.

**SAU\_REGION6.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region6 at reset.

**SAU\_REGION6.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region6 at reset.

**SAU\_REGION6.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region6 at reset.

**SAU\_REGION6.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region6 at reset.

**SAU\_REGION7.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of SAU region7 at reset.

**SAU\_REGION7.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Enable SAU region7 at reset.

**SAU\_REGION7.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of SAU region7 at reset.

**SAU\_REGION7.NSC****Type**

bool

**Default value**

0x0

**Description**

Set NSC for SAU region7 at reset.

**SECEXT****Type**

bool

**Default value**

0x1

**Description**

Whether the ARMv8-M Security Extensions are included.

**WIC****Type**

bool



**Default value**

0x1

**Description**

Include support for WIC-mode deep sleep.

**WICLINES****Type**

int

**Default value**

0x23

**Description**

Number of lines supported by the WIC interface.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**`master_id`****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**`min_sync_level`****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base****Type**

int

**Default value**

0x20800000

**Description**  
Virtual address of base of descending stack.

**semihosting-stack\_limit**

**Type**  
int

**Default value**  
0x20700000

**Description**  
Virtual address of stack limit.

**vfp-enable\_at\_reset**

**Type**  
bool

**Default value**  
0x0

**Description**  
Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

3.5.59 ARMCortexM52CT

CortexM52CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-420: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexM52CT

This model supports the M-Profile Vector Extension (MVE) and the Custom Datapath Extension (CDE). For more information, see [CDE](#).

Iris and MTI instances for ARMCortexM52CT

This model has the following Iris instances:

Table 3-421: ARMCortexM52CT Iris instances

InstanceName	ComponentName
ARMCortexM52CT	ARM_Cortex-M52

InstanceName	ComponentName
ARMCortexM52CT.acp_mapper	PVBusMapper
ARMCortexM52CT.ext_bus	PVBusLogger
ARMCortexM52CT.ext_bus.mapper	PVBusMapper
ARMCortexM52CT.l1_incoherent_interconnect	PVCache
ARMCortexM52CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexM52CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexM52CT.l1dcache	PVCache
ARMCortexM52CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM52CT.l1icache	PVCache
ARMCortexM52CT.l1icache.upstream[0]	PVBusSlave
ARMCortexM52CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-422: ARMCortexM52CT MTI instances**

InstanceName	ComponentName
ARMCortexM52CT	<a href="#">ARM_Cortex-M52</a>
ARMCortexM52CT.acp_mapper	<a href="#">PVBusMapper</a>
ARMCortexM52CT.ext_bus	<a href="#">PVBusLogger</a>
ARMCortexM52CT.ext_bus.mapper	<a href="#">PVBusMapper</a>
ARMCortexM52CT.l1_incoherent_interconnect	<a href="#">PVCache</a>
ARMCortexM52CT.l1_incoherent_interconnect.upstream[0]	<a href="#">PVBusSlave</a>
ARMCortexM52CT.l1_incoherent_interconnect.upstream[10]	<a href="#">PVBusSlave</a>
ARMCortexM52CT.l1_incoherent_interconnect.upstream[11]	<a href="#">PVBusSlave</a>

InstanceName	ComponentName
ARMCortexM52CT.11_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexM52CT.11_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexM52CT.11dcache	PVCache
ARMCortexM52CT.11dcache.upstream[0]	PVBusSlave
ARMCortexM52CT.11icache	PVCache
ARMCortexM52CT.11icache.upstream[0]	PVBusSlave
ARMCortexM52CT.12_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM52CT

**Table 3-423: Ports**

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
ahbp_m	PVBus	Master	The core will generate Vendor System data accesses on this port.
ahbs	PVBus	Slave	Slave AHB - External master (e.g. DMA) can write to TCMs (whether or not enabled in xTCMCR)
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	CoprocBusProtocol	Slave	Co-Processor Interface
core_dside_bus_gasket_in	PVBus	Slave	-
core_dside_bus_gasket_out	PVBus	Master	-
cpuwait	Signal	Slave	Stall the CPU out of reset
currns	Signal	Master	Current Security state of the processor
currpri	Value	Master	Current execution priority.

Name	Protocol	Type	Description
dbgen	Signal	Slave	Invasive debug enable
dbgrestart	Signal	Slave	Request for synchronised exit from halt mode
dbgrestarted	Signal	Master	Handshakes with DBGRESTART
edbgrq	Signal	Slave	External request to enter halt mode
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	Value	Master	Port that sends the value of the FPxxC exception flags (FPIX, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	Signal	Master	Indicates that the processor is in halt mode
idau	PVBus	Master	The core will generate IDAU requests on this port.
idau_invalidate_region	Value_64	Slave	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
initnsvtor	Value	Slave	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initpahben	Signal	Slave	Enable P-AHB on the next reset
initsvtor	Value	Slave	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
inittcmem[2]	Signal	Slave	Reset configuration port - TCM enable initialisation out of reset Bit[0] HIGH = ITCM is enabled Bit[1] HIGH = DTCM is enabled
intnum	Value	Master	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
irq[480]	Signal	Slave	This signal array delivers signals to the NVIC.
lockdcaic	Signal	Slave	Disable access to instruction cache direct cache access registers
lockdtgu	Signal	Slave	Disable writes to registers associated with the DTGU
lockitgu	Signal	Slave	Disable writes to registers associated with the ITGU
locknsmpu	Signal	Slave	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	Signal	Slave	Cortex-M52-specific LOCKNSVTOR, LOCKSVTAIRCR, LOCKSMPU, LOCKNSMPU, LOCKSAU. Disable writes to VTOR_NS
lockpahb	Signal	Slave	P-AHB related ports Disable writes to PAHBCR
locksau	Signal	Slave	Disable writes to the SAU_* registers
locksmpu	Signal	Slave	Disable writes to the Secure MPU_* registers
locksvtaircr	Signal	Slave	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
locktcm	Signal	Slave	Disable writes to registers associated with the TCM
lockup	Signal	Master	Asserted when the processor is in lockup state.
niden	Signal	Slave	Non-invasive debug enable
nmi	Signal	Slave	Configure non maskable interrupt.

Name	Protocol	Type	Description
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
prequest	PChannel	Slave	Low Power Interface
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	PVBus	Master	The core will generate bus requests on this port.
qrequest	PChannel	Slave	-
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
spiden	Signal	Slave	Secure invasive debug enable
spniden	Signal	Slave	Secure non-invasive debug enable
stcalib[2]	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	Signal	Master	Acknowledge signal for WICENREQ
wicenreq	Signal	Slave	Request for deep sleep to be WIC-based deep sleep.
wicsense[483]	Signal	Master	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARMCortexM52CT

### CDEMAPPEDONCP

#### Type

int

#### Default value

0xff

#### Description

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

### CDERTLID

#### Type

int

#### Default value

0x20

#### Description

Value of ID\_AFR0.CDERTLID.



**CFGBIGEND****Type**

bool

**Default value**

0x0

**Description**

Initialize processor to big endian mode.

**CFGCPUINST****Type**

int

**Default value**

0x0

**Description**

CPU instance number. This is part of the TCM base address, in bits 25:24.

**CFGDTCMSZ****Type**

int

**Default value**

0x9

**Description**

Size of the data TCM. 0=No DTCM implemented. Otherwise=Size of DTCM=pow(2, CFGDTCMSZ - 1) KB. Minimum size is 4KB.

**CFGITCMSZ****Type**

int

**Default value**

0x9

**Description**

Size of the instruction TCM. 0=No ITCM implemented. Otherwise=Size of ITCM=pow(2, CFGITCMSZ - 1) KB. Minimum size is 4KB.

**CFGMEMALIAS****Type**

int

**Default value**

0x0

**Description**

Memory address alias bit for the ITCM, DTCM and P-AHB regions. 0=No alias, 1=Alias bit 26, 2=Alias bit 27, 4=Alias bit 28.

**CFGNOCECP****Type**

int

**Default value**

0x0

**Description**

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

**CFGPAACBTI****Type**

bool

**Default value**

0x0

**Description**

Enables support for the Pointer Authentication and Branch Target Identification (PACBTI) Extension. FALSE: Disabled, TRUE:PAC implemented using the QARMA3 algorithm with BTI.

**CFGPAHBSZ****Type**

int

**Default value**

0x0

**Description**

Size of the P-AHB peripheral port memory region. 0=P-AHB disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

**CPIF****Type**

bool

**Default value**

0x1

**Description**

Specifies whether the external coprocessor interface is included.

**CPNSPRESENT****Type**

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

**CPSPPRESENT****Type**

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

**CTI****Type**

bool

**Default value**

0x0

**Description**

CTI (Cross Trigger Interface) included.

**CTI\_irq0\_pin****Type**

int

**Default value**

0x4

**Description**

CTI interrupt request 0 pin.

**CTI\_irq1\_pin****Type**

int

**Default value**

0x5

**Description**

CTI interrupt request 1 pin.

**DBGLVL****Type**

int

**Default value**

0x2

**Description**

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators; 3: 8 Watchpoints, 8 Breakpoint comparators.

**DCACHESZ****Type**

int

**Default value**

0xf

**Description**

Whether the D-cache is included and, if included, the size of it. Bit 0: 0=No D-cache included, 1=D-cache included. Bits [4:1]: 0x0=4KB D-cache, 0x1=8KB D-cache, 0x3=16KB D-cache, 0x7=32KB D-cache, 0xF=64KB D-cache.

**DTGU****Type**

bool

**Default value**

0x0

**Description**

DTCM Security Gate Unit included.

**DTGUBLKSZ****Type**

int

**Default value**

0x3

**Description**

DTCM gate unit block size. Size=pow(2, DTGUBLKSZ + 5) bytes.

**DTGUMAXBLKS****Type**

int

**Default value**

0x0

**Description**

Maximum number of DTCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{DTGUMAXBLKS})$ .

**ECOREVNUM****Type**

int

**Default value**

0x0

**Description**

ECO Revision number.

**ERRDEVID . NUM****Type**

int

**Default value**

0x1

**Description**

RAS: Number of implemented error record indexes, 0 to 1.

**ETM****Type**

bool

**Default value**

0x1

**Description**

Support for ETM trace. false : No ETM trace included, true: ETM trace included.

**FPMVE****Type**

int

**Default value**

0x5

**Description**

Set whether the model has FP and / or MVE support. 0: No FP and MVE support. 1: FP half and single precision. 2: FP half, single and double precision. 3: MVE integer. 4: FP half and single precision and MVE integer. 5: FP half, single and double precision and MVE floating point.

**ICACHESZ****Type**

int

**Default value**

0x3f

**Description**

Whether the I-cache is included and, if included, the size of it. Bit 0: 0=No I-cache included, 1=I-cache included. Bits [6:1]: 0x0=1KB I-cache (only with unified cache), 0x1=2KB I-cache (only with unified cache), 0x3=4KB I-cache, 0x7=8KB I-cache, 0xF=16KB I-cache, 0x1F=32KB I-cache, 0x3F=64KB I-cache.

**INITNSVTOR****Type**

int

**Default value**

0x0

**Description**

Non-Secure vector-table offset at reset.

**INITPAHBEN****Type**

bool

**Default value**

0x0

**Description**

The P-AHB enable state at reset.

**INITSVTOR****Type**

int

**Default value**

0x0

**Description**

Secure vector-table offset at reset.

**INITTCMEN****Type**

int

**Default value**

0x3

**Description**

The TCM enable state at reset. Bit 0 corresponds to ITCM enable state, bit 1 corresponds to DTCM enable state.

**IRQDIS0****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

**IRQDIS1****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

**IRQDIS10****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

**IRQDIS11****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

**IRQDIS12****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

**IRQDIS13****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

**IRQDIS14****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

**IRQDIS2****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

**IRQDIS3****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

**IRQDIS4****Type**

int



**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

**IRQDIS5****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

**IRQDIS6****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

**IRQDIS7****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

**IRQDIS8****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

**IRQDIS9****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

**IRQLVL****Type**

int

**Default value**

0x3

**Description**

Number of bits of interrupt priority.

**ITGU****Type**

bool

**Default value**

0x0

**Description**

ITCM Security Gate Unit included.

**ITGUBLKSZ****Type**

int

**Default value**

0x3

**Description**

ITCM gate unit block size. Size=pow(2, ITGUBLKSZ + 5) bytes.

**ITGUMAXBLKS****Type**

int

**Default value**

0x0

**Description**

Maximum number of ITCM gate unit blocks. Number of blocks=pow(2, ITGUMAXBLKS).

**ITM****Type**

bool

**Default value**

0x1

**Description**

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

**IWIC****Type**

bool

**Default value**

0x1

**Description**

Include support for Internal Wake-up Interrupt Controller.

**LOCKDTGU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Data TGU registers write.

**LOCKITGU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Instruction TGU registers write.

**LOCKTCM****Type**

bool

**Default value**

0x0

**Description**

Lock down of TCM registers write.

**LOCK\_NS\_MPU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Non-Secure MPU registers write.

**LOCK\_SAU****Type**

bool

**Default value**

0x0

**Description**

Lock down of SAU registers write.

**LOCK\_S\_MPU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Secure MPU registers write.

**MPU\_NS****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

**MPU\_S****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

**NUMIRQ****Type**

int

**Default value**

0x20

**Description**

Number of user interrupts.

**SAU****Type**

int

**Default value**

0x4

**Description**

Number of SAU regions (0 => no SAU).

**SECEXT****Type**

bool

**Default value**

0x1

**Description**

Whether the ARMv8-M Security Extensions are included.

**UCACHE****Type**

bool

**Default value**

0x0

**Description**

Whether the I-cache acts as a unified cache (ICACHESZ is used for the size).

**WICLINES****Type**

int

**Default value**

0x23

**Description**

Number of lines supported by the WIC interface.

**cde\_impl\_name****Type**

string

**Default value**

""

**Description**

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**delay\_faultmask\_update****Type**

bool

**Default value**

0x0

**Description**

Delay FAULTMASK update to context sync.

**delay\_sysreg\_update****Type**

bool

**Default value**

0x0

**Description**

Delay some system register updates (e.g. SHCSR) to context sync.

**ecc\_on****Type**

bool

**Default value**

0x0

**Description**

Enable Error Correcting Code.

**has\_cde****Type**

bool

**Default value**

0x0

**Description**

Enables Custom Datapath Extensions.

**has\_core\_dside\_bus\_gasket****Type**

bool

**Default value**

0x0

**Description**

STL gasket enabled.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**master\_id****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.



**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base****Type**

int

**Default value**

0x20800000

**Description**

Virtual address of base of descending stack.

**semihosting-stack\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of stack limit.

**trace\_style****Type**

int

**Default value**

0x2

**Description**

MVE instruction trace style: Add 16 for [\*\*-] beat trace. Add 32 for tracing IMPLIED LOB instructions. Add 64 to change opcode of implied BF to 0xBF00.

**vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**3.5.60 ARMCortexM55CT**

CortexM55CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-424: IP revisions support**

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Changes in 11.29.19**

Ports added:

- `tramdueeca`
- `tramsceeca`

Parameters added:

- `rse_ecc_support`

**About ARMCortexM55CT**

The model supports Custom Datapath Extension (CDE). For more information, see [CDE](#).

The model does not support the following functionality:

- Cross Trigger Interface (CTI).
- Programmable MBIST controller (PMC-100).
- Error Correcting Code (ECC).
- Q-Channel.

The following interfaces and registers are not modeled:

- ITM and ETM trace and trace synchronization and trigger interface signals.
- Dual-core lock-step operation.
- Interrupt latencies.
- Prefetcher Control Register (PFCR).
- Direct cache access registers.

In the Memory System Control Register (MSCR), only the cache-related bits are supported:

- FORCEWT
- DCACTIVE
- ICACTIVE
- IDCCLEAN

## Differences between the model and the RTL

In hardware, PMU\_CCNTR is an alias of the DWT\_CYCCNT register, so they contain the same values. In the model, PMU\_CCNTR is implemented differently to DWT\_CYCCNT, so they contain different values. The value held in DWT\_CYCCNT is not representative of hardware.

## Implementation of ITM in M-class models

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an ITM\_PACKET\_TYPE field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

## Iris and MTI instances for ARM CortexM55CT

This model has the following Iris instances:

**Table 3-426: ARM CortexM55CT Iris instances**

InstanceName	ComponentName
ARM CortexM55CT	ARM_Cortex-M55
ARM CortexM55CT.acp_mapper	PVBusMapper
ARM CortexM55CT.ext_bus	PVBusLogger
ARM CortexM55CT.ext_bus.mapper	PVBusMapper
ARM CortexM55CT.l1_incoherent_interconnect	PVCache

InstanceName	ComponentName
ARMCortexM55CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexM55CT.l1dcache	PVCache
ARMCortexM55CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM55CT.l1licache	PVCache
ARMCortexM55CT.l1licache.upstream[0]	PVBusSlave
ARMCortexM55CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-427: ARMCortexM55CT MTI instances**

InstanceName	ComponentName
ARMCortexM55CT	ARM_Cortex-M55
ARMCortexM55CT.acp_mapper	PVBusMapper
ARMCortexM55CT.ext_bus	PVBusLogger
ARMCortexM55CT.ext_bus.mapper	PVBusMapper
ARMCortexM55CT.l1_incoherent_interconnect	PVCache
ARMCortexM55CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexM55CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave

InstanceName	ComponentName
ARMCortexM55CT.11_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexM55CT.11_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexM55CT.11dcache	PVCache
ARMCortexM55CT.11dcache.upstream[0]	PVBusSlave
ARMCortexM55CT.11icache	PVCache
ARMCortexM55CT.11icache.upstream[0]	PVBusSlave
ARMCortexM55CT.12_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM55CT

**Table 3-428: Ports**

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
ahbp_m	PVBus	Master	The core will generate Vendor System data accesses on this port.
ahbs	PVBus	Slave	Slave AHB - External master (e.g. DMA) can write to TCMs (whether or not enabled in xTCMCR)
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	CoprocBusProtocol	Slave	Co-Processor Interface
core_dside_bus_gasket_in	PVBus	Slave	-
core_dside_bus_gasket_out	PVBus	Master	-
cpuwait	Signal	Slave	Stall the CPU out of reset
currns	Signal	Master	Current Security state of the processor
currpri	Value	Master	Current execution priority.
dbgen	Signal	Slave	Invasive debug enable
dbgrestart	Signal	Slave	Request for synchronised exit from halt mode
dbgrestarted	Signal	Master	Handshakes with DBGRESTART
edbgrq	Signal	Slave	External request to enter halt mode

Name	Protocol	Type	Description
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	Value	Master	Port that sends the value of the FPxxC exception flags (FPIX, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	Signal	Master	Indicates that the processor is in halt mode
idau	PVBus	Master	The core will generate IDAU requests on this port.
idau_invalidate_region	Value_64	Slave	64 bit number to invalidate IDAU memory range (start_address<<32 end_address)
initnsvtor	Value	Slave	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initpahben	Signal	Slave	Enable P-AHB on the next reset
initsvtor	Value	Slave	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
inittcmem[2]	Signal	Slave	Reset configuration port - TCM enable initialisation out of reset Bit[0] HIGH = ITCM is enabled Bit[1] HIGH = DTCM is enabled
intnum	Value	Master	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
irq[480]	Signal	Slave	This signal array delivers signals to the NVIC.
lockdcaic	Signal	Slave	Disable access to instruction cache direct cache access registers
lockdtgu	Signal	Slave	Disable writes to registers associated with the DTGU
lockitgu	Signal	Slave	Disable writes to registers associated with the ITGU
locknsmpu	Signal	Slave	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	Signal	Slave	Cortex-M55-specific LOCKNSVTOR, LOCKSVTAIRCR, LOCKSMPU, LOCKNSMPU, LOCKSAU. Disable writes to VTOR_NS
lockpahb	Signal	Slave	P-AHB related ports Disable writes to PAHBCR
locksau	Signal	Slave	Disable writes to the SAU_* registers
locksmpu	Signal	Slave	Disable writes to the Secure MPU_* registers
locksvtaircr	Signal	Slave	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMIN
locktcm	Signal	Slave	Disable writes to registers associated with the TCM
lockup	Signal	Master	Asserted when the processor is in lockup state.
niden	Signal	Slave	Non-invasive debug enable
nmi	Signal	Slave	Configure non maskable interrupt.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
prequest	PChannel	Slave	Low Power Interface
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
qrequest	PChannel	Slave	-
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
spiden	Signal	Slave	Secure invasive debug enable
spniden	Signal	Slave	Secure non-invasive debug enable
stcalib[2]	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.
tramdueeca	Value	Master	-
tramsceeca	Value	Master	-
wicenack	Signal	Master	Acknowledge signal for WICENREQ
wicenreq	Signal	Slave	Request for deep sleep to be WIC-based deep sleep.
wicsense[483]	Signal	Master	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARMCortexM55CT

### CDEMAPPEDONCP

#### Type

int

#### Default value

0xff

#### Description

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

### CDERTLID

#### Type

int

#### Default value

0x20

#### Description

Value of ID\_AFR0.CDERTLID.



**CFGBIGEND****Type**

bool

**Default value**

0x0

**Description**

Initialize processor to big endian mode.

**CFGDTCMSZ****Type**

int

**Default value**

0x9

**Description**

Size of the data TCM. 0=No DTCM implemented. Otherwise=Size of DTCM=pow(2, CFGDTCMSZ - 1) KB. Minimum size is 4KB.

**CFGITCMSZ****Type**

int

**Default value**

0x9

**Description**

Size of the instruction TCM. 0=No ITCM implemented. Otherwise=Size of ITCM=pow(2, CFGITCMSZ - 1) KB. Minimum size is 4KB.

**CFGMEMALIAS****Type**

int

**Default value**

0x0

**Description**

Memory address alias bit for the ITCM, DTCM and P-AHB regions. 0=No alias, 1=Alias bit 24, 2=Alias bit 25, 4=Alias bit 26, 8=Alias bit 27, 16=Alias bit 28.

**CFGNOCDECP****Type**

int

**Default value**

0x0

**Description**

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

**CFGPAHBSZ****Type**

int

**Default value**

0x0

**Description**

Size of the P-AHB peripheral port memory region. 0=P-AHB disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

**CPIF****Type**

bool

**Default value**

0x1

**Description**

Specifies whether the external coprocessor interface is included.

**CPNSPRESENT****Type**

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

**CPSPRESENT****Type**

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

**CTI****Type**

bool

**Default value**

0x0

**Description**

CTI (Cross Trigger Interface) included.

**CTI\_irq0\_pin****Type**

int

**Default value**

0x4

**Description**

CTI interrupt request 0 pin.

**CTI\_irq1\_pin****Type**

int

**Default value**

0x5

**Description**

CTI interrupt request 1 pin.

**DBG\_LVL****Type**

int

**Default value**

0x2

**Description**

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators; 3: 8 Watchpoints, 8 Breakpoint comparators.

**DCACHESZ****Type**

int

**Default value**

0xf

**Description**

Whether the D-cache is included and, if included, the size of it. Bit 0: 0=No D-cache included, 1=D-cache included. Bits [4:1]: 0x0=4KB D-cache, 0x1=8KB D-cache, 0x3=16KB D-cache, 0x7=32KB D-cache, 0xF=64KB D-cache.

**DTGU****Type**

bool

**Default value**

0x0

**Description**

DTCM Security Gate Unit included.

**DTGUBLKSZ****Type**

int

**Default value**

0x3

**Description**

DTCM gate unit block size. Size=pow(2, DTGUBLKSZ + 5) bytes.

**DTGUMAXBLKS****Type**

int

**Default value**

0x0

**Description**

Maximum number of DTCM gate unit blocks. Number of blocks=pow(2, DTGUMAXBLKS).

**ECOREVNUM****Type**

int

**Default value**

0x0

**Description**

ECO Revision number.

**ERRDEVID . NUM****Type**

int

**Default value**

0x1

**Description**

RAS: Number of implemented error record indexes, 0 to 1.

**ETM****Type**

bool

**Default value**

0x1

**Description**

Support for ETM trace. false : No ETM trace included, true: ETM trace included.

**FPU****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**ICACHESZ****Type**

int

**Default value**

0xf

**Description**

Whether the I-cache is included and, if included, the size of it. Bit 0: 0=No I-cache included, 1=I-cache included. Bits [4:1]: 0x0=4KB I-cache, 0x1=8KB I-cache, 0x3=16KB I-cache, 0x7=32KB I-cache, 0xF=64KB I-cache.

**INITNSVTOR****Type**

int

**Default value**

0x0

**Description**

Non-Secure vector-table offset at reset.

**INITPAHBEN****Type**

bool

**Default value**

0x0

**Description**

The P-AHB enable state at reset.

**INITSVTOR****Type**

int

**Default value**

0x0

**Description**

Secure vector-table offset at reset.

**INITTCMEN****Type**

int

**Default value**

0x3

**Description**

The TCM enable state at reset. Bit 0 corresponds to ITCM enable state, bit 1 corresponds to DTCM enable state.

**IRQDIS0****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

**IRQDIS1****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

**IRQDIS10****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

**IRQDIS11****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

**IRQDIS12****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

**IRQDIS13****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

**IRQDIS14****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

**IRQDIS2****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

**IRQDIS3****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

**IRQDIS4****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

**IRQDIS5****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

**IRQDIS6****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

**IRQDIS7****Type**

int



**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

**IRQDIS8****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

**IRQDIS9****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

**IRQLVL****Type**

int

**Default value**

0x3

**Description**

Number of bits of interrupt priority.

**ITGU****Type**

bool

**Default value**

0x0

**Description**

ITCM Security Gate Unit included.

**ITGUBLKSZ****Type**

int

**Default value**

0x3

**Description**

ITCM gate unit block size. Size=pow(2, ITGUBLKSZ + 5) bytes.

**ITGUMAXBLKS****Type**

int

**Default value**

0x0

**Description**

Maximum number of ITCM gate unit blocks. Number of blocks=pow(2, ITGUMAXBLKS).

**ITM****Type**

bool

**Default value**

0x1

**Description**

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

**IWIC****Type**

bool

**Default value**

0x1

**Description**

Include support for Internal Wake-up Interrupt Controller.

**LOCKDTGU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Data TGU registers write.

**LOCKITGU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Instruction TGU registers write.

**LOCKTCM****Type**

bool

**Default value**

0x0

**Description**

Lock down of TCM registers write.

**LOCK\_NS\_MPU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Non-Secure MPU registers write.

**LOCK\_SAU****Type**

bool

**Default value**

0x0

**Description**

Lock down of SAU registers write.

**LOCK\_S\_MPU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Secure MPU registers write.

**MPU\_NS****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

**MPU\_S****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

**MVE****Type**

int

**Default value**

0x1

**Description**

Set whether the model has MVE support. If FPU = 0: 0=MVE not included, 1=Integer subset of MVE included. If FPU = 1: 0=MVE not included, 1=Integer subset of MVE included, 2=Integer and half and single precision floating point MVE included.

**NUMIRQ****Type**

int

**Default value**

0x20

**Description**

Number of user interrupts.

**SAU****Type**

int

**Default value**

0x4

**Description**

Number of SAU regions (0 => no SAU).

**SECEXT****Type**

bool

**Default value**

0x1

**Description**

Whether the ARMv8-M Security Extensions are included.

**WICLINES****Type**

int

**Default value**

0x23

**Description**

Number of lines supported by the WIC interface.

**cde\_impl\_name****Type**

string

**Default value**

""

**Description**

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**delay\_faultmask\_update****Type**

bool

**Default value**

0x0

**Description**

Delay FAULTMASK update to context sync.

**delay\_sysreg\_update****Type**

bool

**Default value**

0x0

**Description**

Delay some system register updates (e.g. SHCSR) to context sync.

**ecc\_on****Type**

bool

**Default value**

0x0

**Description**

Enable Error Correcting Code.

**has\_cde****Type**

bool

**Default value**

0x0

**Description**

Enables Custom Datapath Extensions.

**has\_core\_dside\_bus\_gasket****Type**

bool

**Default value**

0x0

**Description**

STL gasket enabled.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**master\_id****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**rse\_ecc\_support****Type**

int

**Default value**

0x0

**Description**

Support for ECC initialization in TCM, following RSE spec. 0=No support, 1=Supported, faults are reported as single bit correctable errors, 2=Supported, faults are reported as double bit uncorrected errors.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.



**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base****Type**

int

**Default value**

0x20800000

**Description**  
Virtual address of base of descending stack.

**semihosting-stack\_limit**

**Type**  
int

**Default value**  
0x20700000

**Description**  
Virtual address of stack limit.

**trace\_style**

**Type**  
int

**Default value**  
0x2

**Description**  
MVE instruction trace style: Add 16 for [\*\*-] beat trace. Add 32 for tracing IMPLIED LOB instructions. Add 64 to change opcode of implied BF to 0xBF00.

**vfp-enable\_at\_reset**

**Type**  
bool

**Default value**  
0x0

**Description**  
Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

3.5.61 ARMCortexM85CT

CortexM85CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-429: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Iris and MTI instances for ARM CortexM85CT

This model has the following Iris instances:

**Table 3-430: ARM CortexM85CT Iris instances**

InstanceName	ComponentName
ARM CortexM85CT	ARM_Cortex-M85
ARM CortexM85CT.acp_mapper	PVBusMapper
ARM CortexM85CT.ext_bus	PVBusLogger
ARM CortexM85CT.ext_bus.mapper	PVBusMapper
ARM CortexM85CT.l1_incoherent_interconnect	PVCache
ARM CortexM85CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARM CortexM85CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARM CortexM85CT.l1dcache	PVCache
ARM CortexM85CT.l1dcache.upstream[0]	PVBusSlave
ARM CortexM85CT.l1licache	PVCache
ARM CortexM85CT.l1licache.upstream[0]	PVBusSlave
ARM CortexM85CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-431: ARM CortexM85CT MTI instances**

InstanceName	ComponentName
ARM CortexM85CT	ARM_Cortex-M85
ARM CortexM85CT.acp_mapper	PVBusMapper
ARM CortexM85CT.ext_bus	PVBusLogger

InstanceName	ComponentName
ARMCortexM85CT.ext_bus.mapper	PVBusMapper
ARMCortexM85CT.l1_incoherent_interconnect	PVCache
ARMCortexM85CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexM85CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexM85CT.l1dcache	PVCache
ARMCortexM85CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM85CT.l1icache	PVCache
ARMCortexM85CT.l1icache.upstream[0]	PVBusSlave
ARMCortexM85CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM85CT

**Table 3-432: Ports**

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
ahbp_m	PVBus	Master	The core will generate Vendor System data accesses on this port.
ahbs	PVBus	Slave	Slave AHB - External master (e.g. DMA) can write to TCMs (whether or not enabled in xTCMCR)
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	CoprocBusProtocol	Slave	Co-Processor Interface

Name	Protocol	Type	Description
core_dside_bus_gasket_in	PVBus	Slave	-
core_dside_bus_gasket_out	PVBus	Master	-
cpuwait	Signal	Slave	Stall the CPU out of reset
currns	Signal	Master	Current Security state of the processor
currpri	Value	Master	Current execution priority.
dbgen	Signal	Slave	Invasive debug enable
dbgrestart	Signal	Slave	Request for synchronised exit from halt mode
dbgrestarted	Signal	Master	Handshakes with DBGRESTART
edbgrq	Signal	Slave	External request to enter halt mode
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	Value	Master	Port that sends the value of the FPxxC exception flags (FPIXC, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	Signal	Master	Indicates that the processor is in halt mode
idau	PVBus	Master	The core will generate IDAU requests on this port.
idau_invalidate_region	Value_64	Slave	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
initnsvtor	Value	Slave	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initpahben	Signal	Slave	Enable P-AHB on the next reset
initsvtor	Value	Slave	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
inittcmem[2]	Signal	Slave	Reset configuration port - TCM enable initialisation out of reset Bit[0] HIGH = ITCM is enabled Bit[1] HIGH = DTCM is enabled
intnum	Value	Master	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
irq[480]	Signal	Slave	This signal array delivers signals to the NVIC.
lockdcaic	Signal	Slave	Disable access to instruction cache direct cache access registers
lockdtgu	Signal	Slave	Disable writes to registers associated with the DTGU
lockitgu	Signal	Slave	Disable writes to registers associated with the ITGU
locknsmpu	Signal	Slave	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	Signal	Slave	Cortex-M85 specific LOCKNSVTOR, LOCKSVTAIRCR, LOCKSMPU, LOCKNSMPU, LOCKSAU. Disable writes to VTOR_NS
lockpahb	Signal	Slave	P-AHB related ports Disable writes to PAHBCCR
locksau	Signal	Slave	Disable writes to the SAU_* registers

Name	Protocol	Type	Description
locksmpu	Signal	Slave	Disable writes to the Secure MPU_* registers
locksvtaircr	Signal	Slave	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
locktcm	Signal	Slave	Disable writes to registers associated with the TCM
lockup	Signal	Master	Asserted when the processor is in lockup state.
niden	Signal	Slave	Non-invasive debug enable
nmi	Signal	Slave	Configure non maskable interrupt.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
prequest	PChannel	Slave	Low Power Interface
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	PVBus	Master	The core will generate bus requests on this port.
qrequest	PChannel	Slave	-
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
spiden	Signal	Slave	Secure invasive debug enable
spniden	Signal	Slave	Secure non-invasive debug enable
stcalib[2]	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	Signal	Master	Acknowledge signal for WICENREQ
wicenreq	Signal	Slave	Request for deep sleep to be WIC-based deep sleep.
wicsense[483]	Signal	Master	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARMCortexM85CT

### CDEMAPPEDONCP

#### Type

int

#### Default value

0xff

#### Description

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

**CDERTLID****Type**

int

**Default value**

0x20

**Description**

Value of ID\_AFR0.CDERTLID.

**CFGBIGEND****Type**

bool

**Default value**

0x0

**Description**

Initialize processor to big endian mode.

**CFGCPUINST****Type**

int

**Default value**

0x0

**Description**

CPU instance number. This is part of the TCM base address, in bits 25:24.

**CFGDTCMSZ****Type**

int

**Default value**

0x9

**Description**

Size of the data TCM. 0=No DTCM implemented. Otherwise=Size of DTCM=pow(2, CFGDTCMSZ - 1) KB. Minimum size is 4KB.

**CFGITCMSZ****Type**

int

**Default value**

0x9

**Description**

Size of the instruction TCM. 0=No ITCM implemented. Otherwise=Size of ITCM=pow(2, CFGITCMSZ - 1) KB. Minimum size is 4KB.

**CFGMEMALIAS****Type**

int

**Default value**

0x0

**Description**

Memory address alias bit for the ITCM, DTCM and P-AHB regions. 0=No alias, 1=Alias bit 24, 2=Alias bit 25, 4=Alias bit 26, 8=Alias bit 27, 16=Alias bit 28.

**CFGNOCDECP****Type**

int

**Default value**

0x0

**Description**

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

**CFGPACBTI****Type**

bool

**Default value**

0x0

**Description**

Enables support for the Pointer Authentication and Branch Target Identification (PACBTI) Extension. FALSE: Disabled, TRUE:PAC implemented using the QARMA3 algorithm with BTI.

**CFGPAHBSZ****Type**

int

**Default value**

0x0

**Description**

Size of the P-AHB peripheral port memory region. 0=P-AHB disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.



**CPIF****Type**

bool

**Default value**

0x1

**Description**

Specifies whether the external coprocessor interface is included.

**CPNSPRESENT****Type**

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

**CPSPPRESENT****Type**

int

**Default value**

0xff

**Description**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

**CTI****Type**

bool

**Default value**

0x0

**Description**

CTI (Cross Trigger Interface) included.

**CTI\_irq0\_pin****Type**

int

**Default value**

0x4

**Description**

CTI interrupt request 0 pin.

**CTI\_irq1\_pin****Type**

int

**Default value**

0x5

**Description**

CTI interrupt request 1 pin.

**DBGLVL****Type**

int

**Default value**

0x2

**Description**

1: 4 Watchpoints, 4 Breakpoint comparators; 2: 8 Watchpoints, 8 Breakpoint comparators.

**DCACHESZ****Type**

int

**Default value**

0xf

**Description**

Whether the D-cache is included and, if included, the size of it. Bit 0: 0=No D-cache included, 1=D-cache included. Bits [4:1]: 0x0=4KB D-cache, 0x1=8KB D-cache, 0x3=16KB D-cache, 0x7=32KB D-cache, 0xF=64KB D-cache.

**DTGU****Type**

bool

**Default value**

0x0

**Description**

DTCM Security Gate Unit included.

**DTGUBLKSZ****Type**

int

**Default value**

0x3

**Description**

DTCM gate unit block size. Size= $\text{pow}(2, \text{DTGUBLKSZ} + 5)$  bytes.

**DTGUMAXBLKS****Type**

int

**Default value**

0x0

**Description**

Maximum number of DTCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{DTGUMAXBLKS})$ .

**ECOREVNUM****Type**

int

**Default value**

0x0

**Description**

ECO Revision number.

**ERRDEVID . NUM****Type**

int

**Default value**

0x1

**Description**

RAS: Number of implemented error record indexes, 0 to 1.

**ETM****Type**

bool

**Default value**

0x1

**Description**

Support for ETM trace. false : No ETM trace included, true: ETM trace included.

**FPU****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**ICACHESZ****Type**

int

**Default value**

0xf

**Description**

Whether the I-cache is included and, if included, the size of it. Bit 0: 0=No I-cache included, 1=I-cache included. Bits [4:1]: 0x0=4KB I-cache, 0x1=8KB I-cache, 0x3=16KB I-cache, 0x7=32KB I-cache, 0xF=64KB I-cache.

**INITNSVTOR****Type**

int

**Default value**

0x0

**Description**

Non-Secure vector-table offset at reset.

**INITPAHBEN****Type**

bool

**Default value**

0x0

**Description**

The P-AHB enable state at reset.

**INITSVTOR****Type**

int

**Default value**

0x0

**Description**

Secure vector-table offset at reset.

**INITTCMEN****Type**

int

**Default value**

0x3

**Description**

The TCM enable state at reset. Bit 0 corresponds to ITCM enable state, bit 1 corresponds to DTCM enable state.

**IRQDIS0****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

**IRQDIS1****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

**IRQDIS10****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

**IRQDIS11****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

**IRQDIS12****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

**IRQDIS13****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

**IRQDIS14****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

**IRQDIS2****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

**IRQDIS3****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

**IRQDIS4****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

**IRQDIS5****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

**IRQDIS6****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

**IRQDIS7****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

**IRQDIS8****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

**IRQDIS9****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

**IRQLVL****Type**

int

**Default value**

0x3

**Description**

Number of bits of interrupt priority.

**ITGU****Type**

bool

**Default value**

0x0

**Description**

ITCM Security Gate Unit included.

**ITGUBLKSZ****Type**

int

**Default value**

0x3

**Description**

ITCM gate unit block size. Size=pow(2, ITGUBLKSZ + 5) bytes.

**ITGUMAXBLKS****Type**

int

**Default value**

0x0

**Description**

Maximum number of ITCM gate unit blocks. Number of blocks=pow(2, ITGUMAXBLKS).



**ITM****Type**

bool

**Default value**

0x1

**Description**

Level of instrumentation trace supported. false : No ITM trace included, true: ITM trace included.

**IWIC****Type**

bool

**Default value**

0x1

**Description**

Include support for Internal Wake-up Interrupt Controller.

**LOCKDTGU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Data TGU registers write.

**LOCKITGU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Instruction TGU registers write.

**LOCKTCM****Type**

bool

**Default value**

0x0

**Description**

Lock down of TCM registers write.

**LOCK\_NS\_MPU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Non-Secure MPU registers write.

**LOCK\_SAU****Type**

bool

**Default value**

0x0

**Description**

Lock down of SAU registers write.

**LOCK\_S\_MPU****Type**

bool

**Default value**

0x0

**Description**

Lock down of Secure MPU registers write.

**MPU\_NS****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

**MPU\_S****Type**

int

**Default value**

0x8

**Description**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

**MVE****Type**

int

**Default value**

0x1

**Description**

Set whether the model has MVE support. If FPU = 0: 0=MVE not included, 1=Integer subset of MVE included. If FPU = 1: 0=MVE not included, 1=Integer subset of MVE included, 2=Integer and half and single precision floating point MVE included.

**NUMIRQ****Type**

int

**Default value**

0x20

**Description**

Number of user interrupts.

**SAU****Type**

int

**Default value**

0x4

**Description**

Number of SAU regions (0 => no SAU).

**SECEXT****Type**

bool

**Default value**

0x1

**Description**

Whether the ARMv8-M Security Extensions are included.

**WICLINES****Type**

int

**Default value**

0x23

**Description**

Number of lines supported by the WIC interface.

**cde\_impl\_name****Type**

string

**Default value**

""

**Description**

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**delay\_faultmask\_update****Type**

bool

**Default value**

0x0

**Description**

Delay FAULTMASK update to context sync.

**delay\_sysreg\_update****Type**

bool

**Default value**

0x0

**Description**

Delay some system register updates (e.g. SHCSR) to context sync.

**ecc\_on****Type**

bool

**Default value**

0x0

**Description**

Enable Error Correcting Code.

**has\_cde****Type**

bool

**Default value**

0x0

**Description**

Enables Custom Datapath Extensions.

**has\_core\_dside\_bus\_gasket****Type**

bool

**Default value**

0x0

**Description**

STL gasket enabled.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**master\_id****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base****Type**

int

**Default value**

0x20800000

**Description**

Virtual address of base of descending stack.

**semihosting-stack\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of stack limit.

**trace\_style****Type**

int

**Default value**

0x2



**Description**

MVE instruction trace style: Add 16 for `[**--]` beat trace. Add 32 for tracing IMPLIED LOB instructions. Add 64 to change opcode of implied BF to 0xBF00.

**vfp-enable\_at\_reset**

**Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

3.5.62 **ARMCortexR4CT**

ARMCortexR4CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-433: IP revisions support

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexR4CT**

- The model implements the `cfgie` port, although it is optional in hardware.
- `pvbuss_s` is the slave port to access the TCM RAM. Bits [3:0] of the user flags in the transaction are used to select the TCM:

1

selects the ATCM.

2

selects the BTCM.

Any other value is reserved.
- When you use the `semihosting-cmd_line` parameter to set the command line that is available to semihosting SVC calls, the value of `argv[0]` is set to the first command-line argument, not to the name of an image.

**Differences between the model and the RTL**

This component has the following differences from the corresponding revision of the RTL implementation:

- There is a single memory port combining instruction, data, DMA and peripheral access.
- ECC and parity schemes are not supported (although the registers might be present).
- The dual core redundancy configuration is not supported.
- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- The hardware refers to the TCMs as “A” and “B”. The model refers to these as “i” and “d”.
- The RTL permits two data TCMs, B0 and B1, to be configured for extra bandwidth. These are not modeled.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems.

### Vectored Interrupt Controller (VIC) ports

The ARMCortexR4CT and ARMCortexR5xnCT models implement a simplified model of the Vectored Interrupt Controller (VIC) port.

The protocol consists of two ports:

- The `vic_addr` port signals the vectored interrupt address from the external VIC.
- The `vic_ack` port signals the VIC that an interrupt has been detected and is being serviced.

The expected interrupt sequence is:

1. The software enables the VIC interface by setting the VE bit in the CP15 control register and setting up suitable interrupt routines.
2. The VIC asserts IRQ.
3. Some time later, the processor detects and responds to the IRQ by asserting `vic_ack`.
4. The VIC writes the vector address to the processor using `vic_addr`.
5. The processor de-asserts `vic_ack`.
6. The processor transfers control to the vectored address returned from the VIC.

The interaction between the processor and the VIC is untimed after the processor acknowledges the interrupt, so certain interrupt sequences cannot occur in the code translation processor models.

### Iris and MTI instances for ARMCortexR4CT

This model has the following Iris instances:

**Table 3-434: ARMCortexR4CT Iris instances**

InstanceName	ComponentName
ARMCortexR4CT	ARM_Cortex-R4
ARMCortexR4CT.acp_mapper	PVBusMapper
ARMCortexR4CT.cpu0.l1dcache	PVCache
ARMCortexR4CT.cpu0.l1dcache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexR4CT.cpu0.l1licache	PVCache
ARMCortexR4CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR4CT.ext_bus	PVBusLogger
ARMCortexR4CT.ext_bus.mapper	PVBusMapper
ARMCortexR4CT.l1_incoherent_interconnect	PVCache
ARMCortexR4CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexR4CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-435: ARMCortexR4CT MTI instances**

InstanceName	ComponentName
ARMCortexR4CT	ARM_Cortex-R4
ARMCortexR4CT.acp_mapper	PVBusMapper
ARMCortexR4CT.cpu0.l1dcache	PVCache
ARMCortexR4CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR4CT.cpu0.l1licache	PVCache
ARMCortexR4CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR4CT.ext_bus	PVBusLogger
ARMCortexR4CT.ext_bus.mapper	PVBusMapper
ARMCortexR4CT.l1_incoherent_interconnect	PVCache
ARMCortexR4CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR4CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave

InstanceName	ComponentName
ARMCortexR4CT.11_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexR4CT.11_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexR4CT.12_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexR4CT

**Table 3-436: Ports**

Name	Protocol	Type	Description
cfgend0	Signal	Slave	Configure BE8 mode after a reset.
cfgie	Signal	Slave	Configure big endian instruction format after a reset.
cfgnmfi	Signal	Slave	Configure FIQs as non-maskable after a reset.
cfgte	Signal	Slave	Configure exceptions to be taken in thumb mode after a reset.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cpuhalt	Signal	Slave	Raising this signal will put the core into halt mode.
fiq	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
initramd	Signal	Slave	Configure DTCM enabled after a reset.
initrami	Signal	Slave	Configure ITCM enabled after a reset.
irq	Signal	Slave	This signal drives the CPU's interrupt handling.
loczrama	Signal	Slave	Location of ATCM at reset.
pmuirq	Signal	Master	Interrupt signal from performance monitoring unit.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
pvbus_s	PVBus	Slave	Slave access to TCMs.
reset	Signal	Slave	Raising this signal will put the core into reset mode.
standbywfi	Signal	Master	Signal from the core that it is waiting in standby for an interrupt.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.

Name	Protocol	Type	Description
vic_ack	Signal	Master	Vic acknowledge port to primary VIC.
vic_addr	Value	Slave	Vic address port from primary VIC.
vinithi	Signal	Slave	Configure high vectors after a reset.

## Parameters for ARM Cortex R4CT

### CFGEND0

#### Type

bool

#### Default value

0x0

#### Description

Initialize to BE8 endianness.

### CFGIE

#### Type

bool

#### Default value

0x0

#### Description

Set the reset value of the instruction endian bit.

### CFGNMF1

#### Type

bool

#### Default value

0x0

#### Description

Enable nonmaskable FIQ interrupts on startup.

### CFGTE

#### Type

bool

#### Default value

0x0

#### Description

Initialize to take exceptions in T32 state. Model starts in T32 state.

**INITRAMD****Type**

bool

**Default value**

0x0

**Description**

Set or reset the INITRAMD signal.

**INITRAMI****Type**

bool

**Default value**

0x0

**Description**

Set or reset the INITRAMI signal.

**LOCZRAMI****Type**

bool

**Default value**

0x0

**Description**

Set or reset the LOCZRAMI signal.

**NUM\_MPU\_REGION****Type**

int

**Default value**

0x8

**Description**

Number of MPU regions.

**VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**dcache-size****Type**

int

**Default value**

0x10000

**Description**

Set D-cache size in bytes.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dtcm0\_base****Type**

int

**Default value**

0x800000

**Description**

Base address of DTCM at startup.

**dttcm0\_size****Type**

int

**Default value**

0x8

**Description**

Size of DTCM in KB.

**icache-size****Type**

int

**Default value**

0x10000

**Description**

Set I-cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**implements\_vfp****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has been built with VFP support.

**itcm0\_base****Type**

int

**Default value**

0x0

**Description**

Base address of ITCM at startup.



**itcm0\_size****Type**

int

**Default value**

0x8

**Description**

Size of ITCM in KB.

**master\_id****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_fp\_revision****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**scheduler\_mode****Type**

int

**Default value**

0x0

**Description**

Control the interleaving of instructions in this processor. 0, default long quantum. 1, low latency mode, short quantum and signal checking. 2, lock-breaking mode, long quantum with additional context switches near load-exclusive instructions. WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

**semihosting-ARM\_HLT****Type**

int

**Default value**

0xf000

**Description**

ARM HLT number for semihosting.

**semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

ARM SVC number for semihosting.

**semihosting-Thumb\_HLT****Type**

int

**Default value**

0x3c

**Description**

Thumb HLT number for semihosting.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

Thumb SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**semihosting-hlt-enable****Type**

bool

**Default value**

0x0

**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**semihosting-stack\_limit**

**Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**vfp-enable\_at\_reset**

**Type**

bool

**Default value**

0x0

**Description**

Enable coprocessor access and VFP at reset.

**3.5.63 ARMCortexR5x1CT**

ARMCortexR5x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-437: IP revisions support**

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexR5x1CT**

An ARMCortexR5x2CT component also exists.

The per-core parameters are preceded by cpun., where n identifies the core (0 or 1).

The allowed values for the LOCK\_STEP parameter are:

**0**

Disable. Set for two independent cores.

**1**

Lock Step. Appears to the system as two cores but is internally modeled as a single core.

**3**

Split Lock. Appears to the system as two cores but can be statically configured from reset either as two independent cores or two locked cores. For the model, these are equivalent to Disable and Lock Step, respectively, except for the value of build options registers. The model does not support dynamically splitting and locking the cluster.

`pvbus_s` is the slave port to access the TCM RAM of CPU *n*. Bits [3:0] of the user flags in the transaction are used to select the TCM:

**1**

Selects the ATCM of CPU 0

**2**

Selects the BTCM of CPU 0

**3**

Selects the ATCM of CPU 1

**4**

Selects the BTCM of CPU 1

Any other value is reserved.

## Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- The RR bit in the SCTLR is ignored.
- The Low Latency Peripheral Port is not modeled.
- The model only has a single bus master port combining instruction, data, DMA and peripheral accesses. The CP15 control registers associated with peripheral buses preserve values but do not have any other effect.
- The model only supports static split lock and not dynamic split lock. Contact Arm for details.
- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- The model cannot experience an ECC error and does not support fault injection into the system, so Arm does not provide the ability to set error schemes for the caches or TCMs. Contact Arm if you require a particular value in the Build Options registers.

## Vectored Interrupt Controller (VIC) ports

The ARMCortexR4CT and ARMCortexR5xnCT models implement a simplified model of the Vectored Interrupt Controller (VIC) port.

The protocol consists of two ports:

- The `vic_addr` port signals the vectored interrupt address from the external VIC.

- The `vic_ack` port signals the VIC that an interrupt has been detected and is being serviced.

The expected interrupt sequence is:

1. The software enables the VIC interface by setting the VE bit in the CP15 control register and setting up suitable interrupt routines.
2. The VIC asserts IRQ.
3. Some time later, the processor detects and responds to the IRQ by asserting `vic_ack`.
4. The VIC writes the vector address to the processor using `vic_addr`.
5. The processor de-asserts `vic_ack`.
6. The processor transfers control to the vectored address returned from the VIC.

The interaction between the processor and the VIC is untimed after the processor acknowledges the interrupt, so certain interrupt sequences cannot occur in the code translation processor models.

### Iris and MTI instances for ARMCortexR5x1CT

This model has the following Iris instances:

**Table 3-438: ARMCortexR5x1CT Iris instances**

InstanceName	ComponentName
ARMCortexR5x1CT	Cluster_ARM_Cortex-R5
ARMCortexR5x1CT.acp_mapper	PVBusMapper
ARMCortexR5x1CT.cpu0	ARM_Cortex-R5
ARMCortexR5x1CT.cpu0.l1dcache	PVCache
ARMCortexR5x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR5x1CT.cpu0.l1icache	PVCache
ARMCortexR5x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR5x1CT.ext_bus	PVBusLogger
ARMCortexR5x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR5x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave

InstanceName	ComponentName
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexR5x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-439: ARMCortexR5x1CT MTI instances**

InstanceName	ComponentName
ARMCortexR5x1CT.acp_mapper	PVBusMapper
ARMCortexR5x1CT.cpu0	ARM_Cortex-R5
ARMCortexR5x1CT.cpu0.l1dcache	PVCache
ARMCortexR5x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR5x1CT.cpu0.l1icache	PVCache
ARMCortexR5x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR5x1CT.ext_bus	PVBusLogger
ARMCortexR5x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR5x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave



InstanceName	ComponentName
ARMCortexR5x1CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexR5x1CT

**Table 3-440: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	ACP slave port.
cfgatcmsz[2]	Value	Slave	ATCM size.
cfgbtcmsz[2]	Value	Slave	BTCM Size.
cfgend[2]	Signal	Slave	This signal is for EE bit initialisation. This is CFGEE in RTL but cfgend here fastsim consistency reasons.
cfgnmfi[2]	Signal	Slave	Controls non-maskable Fast Interrupts.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cpuhalt[2]	Signal	Slave	Raising this signal will put the core into halt mode.
event[2]	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[2]	Signal	Slave	This signal drives the CPU's fast-interrupt handling.
groupid	Value	Slave	Group ID used for MPIDR.
initrama[2]	Signal	Slave	If ATCM is enabled at reset.
initramb[2]	Signal	Slave	If BTCM is enabled at reset.
irq[2]	Signal	Slave	This signal drives the CPU's interrupt handling.
loczrama[2]	Signal	Slave	Location of ATCM at reset.
pmuirq[2]	Signal	Master	Interrupt signal from performance monitoring unit.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
pvbus_s[1]	PVBus	Slave	tcm slave port.
reset[2]	Signal	Slave	Raising this signal will put the core into reset mode.
standbywfe[2]	Signal	Master	This signal indicate if a core is in wfe state RTL calls this WFEPIPESTOPPED.
standbywfi[2]	Signal	Master	This signal indicates if a core is in WFI state RTL uses WFIPIPESTOPPED.
teinit[2]	Signal	Slave	Default exception handling state.
ticks[2]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vic_ack[2]	Signal	Master	Vic acknowledge port to primary VIC.
vic_addr[2]	Value	Slave	Vic address port from primary VIC.
vinithi[2]	Signal	Slave	This signal controls of the location of the exception vectors at reset.

## Parameters for ARMCortexR5x1CT

### GROUP\_ID

#### Type

int

**Default value**

0x0

**Description**

Value read in GROUP ID register field, bits[15:8] of the MPIDR.

**INST\_ENDIAN****Type**

bool

**Default value**

0x1

**Description**

Controls whether the model supports the instruction endianness bit.

**LOCK\_STEP****Type**

int

**Default value**

0x0

**Description**

Affects dual-processor configurations only, and ignored by single-processor configurations.

**MICRO\_SCU****Type**

bool

**Default value**

0x1

**Description**

Controls whether the effects of the MicroSCU are modeled.

**NUM\_BREAKPOINTS****Type**

int

**Default value**

0x3

**Description**

Controls with how many breakpoint pairs the model has been configured. This only affects the build options registers, because debug is not modeled.

**NUM\_MPU\_REGION****Type**

int

**Default value**

0xc

**Description**

Sets the number of MPU regions.

**NUM\_WATCHPOINTS****Type**

int

**Default value**

0x2

**Description**

Controls with how many watchpoint pairs the model has been configured. This only affects the build options registers, because debug is not modeled.

**SLSPLIT****Type**

bool

**Default value**

0x0

**Description**

Sets whether the model starts in split mode or locked mode.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGATCMSZ****Type**

int

**Default value**

0xe

**Description**

Sets the size of the ATCM.

**cpuX.CFGBTCMSZ****Type**

int

**Default value**

0xe

**Description**

Sets the size of the BTCM.

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Initialize to BE8 endianness.

**cpuX.CFGIE****Type**

bool

**Default value**

0x0

**Description**

Set the reset value of the instruction endian bit.

**cpuX.CFGNMF1****Type**

bool

**Default value**

0x0

**Description**

Enable nonmaskable FIQ interrupts on startup.

**cpuX.DP\_FLOAT****Type**

bool

**Default value**

0x1

**Description**

Sets whether double-precision instructions are available.

**cpuX.INITRAMA****Type**

bool

**Default value**

0x0

**Description**

Initialize with TCMA enabled.

**cpuX.INITRAMB****Type**

bool

**Default value**

0x0

**Description**

Initialize with TCMB enabled.

**cpuX.LOCZRAMA****Type**

bool

**Default value**

0x0

**Description**

Initialize with LOCZRAMA set to 1.

**cpuX.TEINIT****Type**

bool

**Default value**

0x0

**Description**

T32 exception enable. The default has exceptions including reset handled in A32 state.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**cpuX.atcm\_base****Type**

int

**Default value**

0x40000000

**Description**

Model-specific. Sets the base address of the ATCM (forced to 0 if LOCZRAMA is 1).

**cpuX.btcn\_base****Type**

int

**Default value**

0x0

**Description**

Model-specific. Sets the base address of the BTCN (forced to 0 if LOCZRAMA is 0).

**cpuX.dcache-size****Type**

int

**Default value**

0x10000

**Description**

Set D-cache size in bytes.

**cpuX.icache-size****Type**

int

**Default value**

0x10000

**Description**

Set I-cache size in bytes.

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-ARM\_HLT****Type**

int

**Default value**

0xf000

**Description**

ARM HLT number for semihosting.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

ARM SVC number for semihosting.

**cpuX.semihosting-Thumb\_HLT****Type**

int

**Default value**

0x3c

**Description**

Thumb HLT number for semihosting.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

Thumb SVC number for semihosting.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int



**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-hlt-enable****Type**

bool

**Default value**

0x0

**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable coprocessor access and VFP at reset.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether model has VFP support.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**reported\_fp\_revision****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**scheduler\_mode****Type**

int

**Default value**

0x0

**Description**

Control the interleaving of instructions in this processor. 0, default long quantum. 1, low latency mode, short quantum and signal checking. 2, lock-breaking mode, long quantum with additional context switches near load-exclusive instructions. **WARNING:** This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

## 3.5.64 ARM Cortex R7x1CT

ARM Cortex R7x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-441: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About ARM Cortex-R7x1CT

An ARM Cortex-R7x2CT component also exists.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0 or 1).

`pvtbus_s` is the slave port to access the TCM RAM of CPU `n`. Bits [3:0] of the user flags in the transaction are used to select the TCM:

**0**

Selects the ITCM of CPU 0

**1**

Selects the DTCM of CPU 0

**2**

Selects the ITCM of CPU 1

**3**

Selects the DTCM of CPU 1

Any other value is reserved.

When you use the `semihosting-cmd_line` parameter to set the command line that is available to semihosting SVC calls, the value of `argv[0]` is set to the first command-line argument, not to the name of an image.

### Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- This component does not implement address filtering within the SCU. The enable bit for this feature is ignored.
- The GIC does not respect the `CFGSDISABLE` signal. This leads to some registers being accessible when they must not be.
- The SCU enable bit is ignored. The SCU is always enabled.
- The SCU ignores the invalidate all register.
- The Broadcast TLB or cache operations in this model do not cause other cores in the cluster that are asleep because of WFI to wake up.
- The RR bit in the SCTL is ignored.
- The Power Control Register in the system control coprocessor is implemented but writing to it does not change the behavior of the model.

- The model cannot be configured with a 128-entry TLB.
- When modeling the SCU, coherency operations are represented by a memory write followed by a read to refill from memory, rather than using cache-to-cache transfers.
- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- This component implements L1 cache as architecturally defined, but does not implement L2 cache. If you require an L2 cache you can add a PL310 Level 2 Cache Controller component.
- ECC and parity schemes are hardware-specific so are not supported.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems.

### Iris and MTI instances for ARMCortexR7x1CT

This model has the following Iris instances:

**Table 3-442: ARMCortexR7x1CT Iris instances**

InstanceName	ComponentName
ARMCortexR7x1CT	Cluster_ARM_Cortex-R7
ARMCortexR7x1CT.ARM_CortexR7x1CT.debug_rom	debug_rom
ARMCortexR7x1CT.acp_mapper	PVBusMapper
ARMCortexR7x1CT.cpu0	ARM_Cortex-R7
ARMCortexR7x1CT.cpu0.l1dcache	PVCache
ARMCortexR7x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR7x1CT.cpu0.l1icache	PVCache
ARMCortexR7x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR7x1CT.ext_bus	PVBusLogger
ARMCortexR7x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR7x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave

InstanceName	ComponentName
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexR7x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-443: ARMCortexR7x1CT MTI instances**

InstanceName	ComponentName
ARMCortexR7x1CT.acp_mapper	PVBusMapper
ARMCortexR7x1CT.cpu0	ARM_Cortex-R7
ARMCortexR7x1CT.cpu0.l1dcache	PVCache
ARMCortexR7x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR7x1CT.cpu0.l1icache	PVCache
ARMCortexR7x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR7x1CT.ext_bus	PVBusLogger
ARMCortexR7x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR7x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexR7x1CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexR7x1CT

Table 3-444: Ports

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
cfgend[1]	Signal	Slave	This signal is for EE bit initialisation.
cfgnmfi[1]	Signal	Slave	This signal disables FIQ mask in CPSR.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[1]	Signal	Slave	Legacy FIQ request input line.
fiqout[1]	Signal	Master	Output of individual processor nFIQ from the interrupt controller.
fpuflags[1]	ValueState	Master	Floating-Point Unit output flags.
halt[1]	Signal	Slave	Raising this signal will put the core into halt mode. Equivalent to the hardware nCPUHALT[N:0] signal.
initram[1]	Signal	Slave	This signal enables the processor to boot from the instruction TCM.
ints[480]	Signal	Slave	Interrupt distributor interrupt lines.
irq[1]	Signal	Slave	Legacy IRQ request input line.
irqout[1]	Signal	Master	Output of individual processor nIRQ from the interrupt controller.
mfilteren	Signal	Slave	This signal enables filtering of address ranges between master bus ports.
mfilterend	Value	Slave	This port sets end of region mapped to pvbus_m1.
mfilterstart	Value	Slave	This port sets start of region mapped to pvbus_m1.
periphbase	Value	Slave	This port sets the base address of private peripheral region.
periphclk_in	ClockSignal	Slave	The timer and the watchdog take need a clk that is scaled down at least by factor of two.
periphreset	Signal	Slave	This signal resets timer and interrupt controller.
pfilterend	Value	Slave	This port sets end of region mapped to pvbus_mp.
pfilterstart	Value	Slave	This port sets start of region mapped to pvbus_mp.
pmuirq[2]	Signal	Master	Interrupt signal from performance monitoring unit.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_m1	PVBus	Master	The core will generate bus requests on this port.
pvbus_mp	PVBus	Master	The core will generate bus requests on this port.
pvbus_s	PVBus	Slave	tcm slave port
reset[1]	Signal	Slave	Raising this signal will put the core into reset mode.
scureset	Signal	Slave	This signal resets SCU.
smpnamp[1]	Signal	Master	This signals AMP or SMP mode for each Cortex-R7 processor.
standbywfe[1]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[1]	Signal	Master	This signal indicates if a core is in WFI state.
teinit[1]	Signal	Slave	This signal provides default exception handling state.
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.

Name	Protocol	Type	Description
vinithi[1]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
wdreset[1]	Signal	Slave	This signal resets individual watchdog.
wdresetreq[1]	Signal	Master	CPU watchdog reset requests.

## Parameters for ARM Cortex R7x1CT

### CLUSTER\_ID

#### Type

int

#### Default value

0x0

#### Description

Processor cluster ID value.

### LOCK\_STEP

#### Type

int

#### Default value

0x0

#### Description

Affects dual-processor configurations only, and ignored by single-processor configurations. 0 - Disable. Set for two independent processors. 1 - Lock Step. Appears to the system as two processors but is internally modeled as a single processor. 3 - Split Lock. Appears to the system as two processors but can be statically configured from reset either as two independent processors or two locked processors. For the model, these are equivalent to Disable and Lock Step, respectively, except for the value of build options registers. The model does not support dynamically splitting and locking the processor.

### MFILTEREN

#### Type

bool

#### Default value

0x0

#### Description

Enables filtering of address ranges.

### MFILTEREND

#### Type

int



**Default value**

0x0

**Description**

Specifies the end address for address filtering.

**MFILTERSTART****Type**

int

**Default value**

0x0

**Description**

Specifies the start address for address filtering.

**NUM\_MPU\_REGION****Type**

int

**Default value**

0xc

**Description**

Sets the number of MPU regions.

**PERIPHBASE****Type**

int

**Default value**

0xae000000

**Description**

Base address of peripheral memory space.

**PFILTEREND****Type**

int

**Default value**

0x0

**Description**

Specifies the end address for peripheral port address filtering.

**PFILTERSTART****Type**

int

**Default value**

0xffff00000

**Description**

Specifies the start address for peripheral port address filtering.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Initialize to BE8 endianness.

**`cpuX.CFGNMFI`****Type**

bool

**Default value**

0x0

**Description**

Enable nonmaskable FIQ interrupts on startup.

**`cpuX.DP_FLOAT`****Type**

bool

**Default value**

0x1

**Description**

Sets whether double-precision instructions are available.

**cpuX . INITRAM****Type**

bool

**Default value**

0x0

**Description**

Enable the processor to boot from the instruction TCM.

**cpuX . POWERCTLI****Type**

int

**Default value**

0x0

**Description**

Default power control state for processor.

**cpuX . SMPnAMP****Type**

bool

**Default value**

0x0

**Description**

Set whether the processor is part of a coherent domain.

**cpuX . TEINIT****Type**

bool

**Default value**

0x0

**Description**

T32 exception enable. The default has exceptions including reset handled in A32 state.

**cpuX . VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**cpuX.dcache-size****Type**

int

**Default value**

0x8000

**Description**

Set D-cache size in bytes.

**cpuX.dtcn\_size****Type**

int

**Default value**

0x8

**Description**

DTCM size in KB.

**cpuX.icache-size****Type**

int

**Default value**

0x8000

**Description**

Set I-cache size in bytes.

**cpuX.itcm\_size****Type**

int

**Default value**

0x8

**Description**

ITCM size in KB.

**cpuX.min\_sync\_level****Type**

int

**Default value**  
0x0

**Description**  
Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-ARM\_HLT**

**Type**  
int

**Default value**  
0xf000

**Description**  
ARM HLT number for semihosting.

**cpuX.semihosting-ARM\_SVC**

**Type**  
int

**Default value**  
0x123456

**Description**  
ARM SVC number for semihosting.

**cpuX.semihosting-Thumb\_HLT**

**Type**  
int

**Default value**  
0x3c

**Description**  
Thumb HLT number for semihosting.

**cpuX.semihosting-Thumb\_SVC**

**Type**  
int

**Default value**  
0xab

**Description**  
Thumb SVC number for semihosting.

**cpuX.semihosting-cmd\_line**

**Type**  
string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-hlt-enable****Type**

bool

**Default value**

0x0

**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.tcm-present****Type**

bool

**Default value**

0x1

**Description**

Disables the DTCM and ITCM.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable coprocessor access and VFP at reset.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether model has VFP support.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dic-spi\_count****Type**

int

**Default value**

0x40

**Description**

Number of shared peripheral interrupts implemented.

**ecc\_on****Type**

bool

**Default value**

0x0



**Description**

Enable Error Correcting Code.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**reported\_fp\_revision****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**scheduler\_mode**

**Type**  
int

**Default value**  
0x0

**Description**  
Control the interleaving of instructions in this processor. 0, default long quantum. 1, low latency mode, short quantum and signal checking. 2, lock-breaking mode, long quantum with additional context switches near load-exclusive instructions. WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

3.5.65 ARMCortexR8x1CT

ARMCortexR8x1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-445: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexR8x1CT

The following components also exist:

ARMCortexR8x2CT. ARMCortexR8x3CT. ARMCortexR8x4CT.

The per-core parameters are preceded by cpun., where n identifies the core (0-3).

pvbus\_s is the slave port to access the TCM RAM of CPU n. Bits [3:0] of the user flags in the transaction are used to select the TCM:

- 0
- Selects the ITCM of CPU 0.
- 1
- Selects the DTCM of CPU 0.
- 2
- Selects the ITCM of CPU 1.
- 3
- Selects the DTCM of CPU 1.
- 4
- Selects the ITCM of CPU 2.

**5**

Selects the DTCM of CPU 2.

**6**

Selects the ITCM of CPU 3.

**7**

Selects the DTCM of CPU 3.

Any other value is reserved.

The `semihosting-cwd` parameter sets the CWD that is used for semihosting. The host operating system limits the maximum path length. The `semihosting-cwd` parameter does not provide any security. Software running on the model can access files outside this directory using relative paths containing `..` or using absolute paths.

## Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- This component does not implement address filtering within the SCU. The enable bit for this feature is ignored.
- The GIC does not respect the `CFGSDISABLE` signal. This leads to some registers being accessible when they must not be.
- The SCU enable bit is ignored. The SCU is always enabled.
- The SCU ignores the invalidate all register.
- The Broadcast TLB or cache operations in this model do not cause other cores in the cluster that are asleep because of WFI to wake up.
- The RR bit in the SCTLR is ignored.
- The Power Control Register in the system control coprocessor is implemented but writing to it does not change the behavior of the model.
- The model cannot be configured with a 128-entry TLB.
- When modeling the SCU, coherency operations are represented by a memory write followed by a read to refill from memory, rather than using cache-to-cache transfers.
- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- This component implements L1 cache as architecturally defined, but does not implement L2 cache. If you require an L2 cache you can add a PL310 Level 2 Cache Controller component.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems.
- ECC and parity schemes are hardware-specific so are not supported.

## Iris and MTI instances for ARMCortexR8x1CT

This model has the following Iris instances:

**Table 3-446: ARMCortexR8x1CT Iris instances**

InstanceName	ComponentName
ARMCortexR8x1CT	Cluster_ARM_Cortex-R8
ARMCortexR8x1CT.ARM_CortexR8x1CT.debug_rom	debug_rom
ARMCortexR8x1CT.acp_mapper	PVBusMapper
ARMCortexR8x1CT.cpu0	ARM_Cortex-R8
ARMCortexR8x1CT.cpu0.l1dcache	PVCache
ARMCortexR8x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR8x1CT.cpu0.l1licache	PVCache
ARMCortexR8x1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR8x1CT.ext_bus	PVBusLogger
ARMCortexR8x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR8x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexR8x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-447: ARMCortexR8x1CT MTI instances**

InstanceName	ComponentName
ARMCortexR8x1CT.acp_mapper	PVBusMapper
ARMCortexR8x1CT.cpu0	ARM_Cortex-R8
ARMCortexR8x1CT.cpu0.l1dcache	PVCache
ARMCortexR8x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexR8x1CT.cpu0.l1icache	PVCache
ARMCortexR8x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR8x1CT.ext_bus	PVBusLogger
ARMCortexR8x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR8x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[10]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[11]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[12]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[13]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[14]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[15]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[16]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[17]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[2]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[3]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[4]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[5]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[6]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[7]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[8]	PVBusSlave
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[9]	PVBusSlave
ARMCortexR8x1CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexR8x1CT

**Table 3-448: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
cfgend[1]	Signal	Slave	This signal is for EE bit initialisation.
cfgnmfi[1]	Signal	Slave	This signal disables FIQ mask in CPSR.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[1]	Signal	Slave	Legacy FIQ request input line.
fiqout[1]	Signal	Master	Output of individual processor nFIQ from the interrupt controller.
fpfilterend0	Value	Slave	This port sets end of region mapped to pvbus_mfp0.
fpfilterend1	Value	Slave	This port sets end of region mapped to pvbus_mfp1.

Name	Protocol	Type	Description
fpfilterend2	Value	Slave	This port sets end of region mapped to pvbus_mfp2.
fpfilterend3	Value	Slave	This port sets end of region mapped to pvbus_mfp3.
fpfilterstart0	Value	Slave	This port sets start of region mapped to pvbus_mfp0.
fpfilterstart1	Value	Slave	This port sets start of region mapped to pvbus_mfp1.
fpfilterstart2	Value	Slave	This port sets start of region mapped to pvbus_mfp2.
fpfilterstart3	Value	Slave	This port sets start of region mapped to pvbus_mfp3.
fpuflags[1]	ValueState	Master	Floating-Point Unit output flags.
halt[1]	Signal	Slave	Raising this signal will put the core into halt mode. Equivalent to the hardware nCPUHALT[N:0] signal.
initram[1]	Signal	Slave	This signal enables the processor to boot from the instruction TCM.
ints[480]	Signal	Slave	Interrupt distributor interrupt lines.
irq[1]	Signal	Slave	Legacy IRQ request input line.
irqout[1]	Signal	Master	Output of individual processor nIRQ from the interrupt controller.
mfilteren	Signal	Slave	This signal enables filtering of address ranges between master bus ports.
mfilterend	Value	Slave	This port sets end of region mapped to pvbus_m1.
mfilterstart	Value	Slave	This port sets start of region mapped to pvbus_m1.
periphbase	Value	Slave	This port sets the base address of private peripheral region.
periphclk_in	ClockSignal	Slave	The timer and the watchdog take need a clk that is scaled down at least by factor of two.
periphreset	Signal	Slave	This signal resets timer and interrupt controller.
pfilterend	Value	Slave	This port sets end of region mapped to pvbus_mp.
pfilterstart	Value	Slave	This port sets start of region mapped to pvbus_mp.
pmuirq[2]	Signal	Master	Interrupt signal from performance monitoring unit.
pvbus_m0	PVBus	Master	AXI master port 0.
pvbus_m1	PVBus	Master	AXI master port 1.
pvbus_mfp0	PVBus	Master	Fast peripheral port for core 0.
pvbus_mfp1	PVBus	Master	Fast peripheral port for core 1.
pvbus_mfp2	PVBus	Master	Fast peripheral port for core 2.
pvbus_mfp3	PVBus	Master	Fast peripheral port for core 3.
pvbus_mp	PVBus	Master	Shared peripheral port.
pvbus_s	PVBus	Slave	tcm slave port.
reset[1]	Signal	Slave	Raising this signal will put the core into reset mode.
scureset	Signal	Slave	This signal resets SCU.
smpnamp[1]	Signal	Master	This signals AMP or SMP mode for each Cortex-R8 processor.
standbywfe[1]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[1]	Signal	Master	This signal indicates if a core is in WFI state.
teinit[1]	Signal	Slave	This signal provides default exception handling state.
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vinithi[1]	Signal	Slave	This signal controls of the location of the exception vectors at reset.

Name	Protocol	Type	Description
wdreset[1]	Signal	Slave	This signal resets individual watchdog.
wdresetreq[1]	Signal	Master	CPU watchdog reset requests.

## Parameters for ARM Cortex R8x1CT

### CLUSTER\_ID

#### Type

int

#### Default value

0x0

#### Description

Processor cluster ID value.

### FPFILTEREND0

#### Type

int

#### Default value

0x0

#### Description

Specifies the end address for the fast peripheral port address filtering.

### FPFILTEREND1

#### Type

int

#### Default value

0x0

#### Description

Specifies the end address for the fast peripheral port address filtering.

### FPFILTEREND2

#### Type

int

#### Default value

0x0

#### Description

Specifies the end address for the fast peripheral port address filtering.

**FPFILTEREND3****Type**

int

**Default value**

0x0

**Description**

Specifies the end address for the fast peripheral port address filtering.

**FPFILTERSTART0****Type**

int

**Default value**

0xffff00000

**Description**

Specifies the start address for the fast peripheral port address filtering.

**FPFILTERSTART1****Type**

int

**Default value**

0xffff00000

**Description**

Specifies the start address for the fast peripheral port address filtering.

**FPFILTERSTART2****Type**

int

**Default value**

0xffff00000

**Description**

Specifies the start address for the fast peripheral port address filtering.

**FPFILTERSTART3****Type**

int

**Default value**

0xffff00000

**Description**

Specifies the start address for the fast peripheral port address filtering.



**LOCK\_STEP****Type**

int

**Default value**

0x0

**Description**

Affects dual-processor configurations only, and ignored by single-processor configurations. 0 - Disable. Set for two independent processors. 1 - Lock Step. Appears to the system as two processors but is internally modeled as a single processor. 3 - Split Lock. Appears to the system as two processors but can be statically configured from reset either as two independent processors or two locked processors. For the model, these are equivalent to Disable and Lock Step, respectively, except for the value of build options registers. The model does not support dynamically splitting and locking the processor.

**MFILTEREN****Type**

bool

**Default value**

0x0

**Description**

Enables filtering of address ranges.

**MFILTEREND****Type**

int

**Default value**

0x0

**Description**

Specifies the end address for address filtering.

**MFILTERSTART****Type**

int

**Default value**

0x0

**Description**

Specifies the start address for address filtering.

**NUM\_MPU\_REGION****Type**

int

**Default value**

0xc

**Description**

Sets the number of MPU regions.

**PERIPHBASE****Type**

int

**Default value**

0xae000000

**Description**

Base address of peripheral memory space.

**PFILTEREND****Type**

int

**Default value**

0x0

**Description**

Specifies the end address for peripheral port address filtering.

**PFILTERSTART****Type**

int

**Default value**

0xffff0000

**Description**

Specifies the start address for peripheral port address filtering.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Initialize to BE8 endianness.

**cpuX.CFGNMFI****Type**

bool

**Default value**

0x0

**Description**

Enable nonmaskable FIQ interrupts on startup.

**cpuX.DP\_FLOAT****Type**

bool

**Default value**

0x1

**Description**

Sets whether double-precision instructions are available.

**cpuX.INITRAM****Type**

bool

**Default value**

0x0

**Description**

Enable the processor to boot from the instruction TCM.

**cpuX . POWERCTLI****Type**

int

**Default value**

0x0

**Description**

Default power control state for processor.

**cpuX . SMPnAMP****Type**

bool

**Default value**

0x0

**Description**

Set whether the processor is part of a coherent domain.

**cpuX . TEINIT****Type**

bool

**Default value**

0x0

**Description**

T32 exception enable. The default has exceptions including reset handled in A32 state.

**cpuX . VINITHI****Type**

bool

**Default value**

0x0

**Description**

Initialize with high vectors enabled.

**cpuX . dcache-size****Type**

int

**Default value**

0x8000

**Description**

Set D-cache size in bytes.

**cpuX.dtcm\_size****Type**

int

**Default value**

0x8

**Description**

DTCM size in KB.

**cpuX.icache-size****Type**

int

**Default value**

0x8000

**Description**

Set I-cache size in bytes.

**cpuX.itcm\_size****Type**

int

**Default value**

0x8

**Description**

ITCM size in KB.

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-ARM\_HLT****Type**

int

**Default value**

0xf000

**Description**

ARM HLT number for semihosting.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

ARM SVC number for semihosting.

**cpuX.semihosting-Thumb\_HLT****Type**

int

**Default value**

0x3c

**Description**

Thumb HLT number for semihosting.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

Thumb SVC number for semihosting.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-hlt-enable****Type**

bool

**Default value**

0x0

**Description**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

**cpuX.semihosting-prefix****Type**

bool

**Default value**

0x0

**Description**

Prefix semihosting output with target instance name.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.tcm-present****Type**

bool

**Default value**

0x1

**Description**

Disables the DTCM and ITCM.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable coprocessor access and VFP at reset.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1



**Description**

Set whether model has VFP support.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dic-spi\_count****Type**

int

**Default value**

0x40

**Description**

Number of shared peripheral interrupts implemented.

**ecc\_on****Type**

bool

**Default value**

0x0

**Description**

Enable Error Correcting Code.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**reported\_fp\_revision****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**scheduler\_mode****Type**

int

**Default value**

0x0

**Description**

Control the interleaving of instructions in this processor. 0, default long quantum. 1, low latency mode, short quantum and signal checking. 2, lock-breaking mode, long quantum with additional context switches near load-exclusive instructions. **WARNING:** This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

## 3.5.66 ARM CortexR52CT

ARM CortexR52CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-449: IP revisions support**

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Changes in 11.29.19

Parameters removed:

- `cpu4.CFGEND`
- `cpu4.CFGTE`
- `cpu4.RVBARADDR`
- `cpu4.ase-present`
- `cpu4.dcache-size`
- `cpu4.enable_trace_special_hlt_imm16`
- `cpu4.flash.enable`
- `cpu4.icache-size`
- `cpu4.llpp.base`
- `cpu4.llpp.size`
- `cpu4.max_code_cache_mb`
- `cpu4.min_sync_level`
- `cpu4.semihosting-A32_HLT`
- `cpu4.semihosting-ARM_SVC`
- `cpu4.semihosting-T32_HLT`
- `cpu4.semihosting-Thumb_SVC`
- `cpu4.semihosting-cmd_line`
- `cpu4.semihosting-cwd`
- `cpu4.semihosting-enable`
- `cpu4.semihosting-heap_base`
- `cpu4.semihosting-heap_limit`
- `cpu4.semihosting-stack_base`
- `cpu4.semihosting-stack_limit`
- `cpu4.tcm.a.enable`
- `cpu4.tcm.a.size`
- `cpu4.tcm.a.wait`
- `cpu4.tcm.b.size`
- `cpu4.tcm.b.wait`

- `cpu4.tcm.c.size`
- `cpu4.tcm.c.wait`
- `cpu4.trace_special_hlt_imm16`
- `cpu4.vfp-dp-present`
- `cpu4.vfp-enable_at_reset`
- `cpu5.CFGEND`
- `cpu5.CFGTE`
- `cpu5.RVBARADDR`
- `cpu5.ase-present`
- `cpu5.dcache-size`
- `cpu5.enable_trace_special_hlt_imm16`
- `cpu5.flash.enable`
- `cpu5.icache-size`
- `cpu5.llpp.base`
- `cpu5.llpp.size`
- `cpu5.max_code_cache_mb`
- `cpu5.min_sync_level`
- `cpu5.semihosting-A32_HLT`
- `cpu5.semihosting-ARM_SVC`
- `cpu5.semihosting-T32_HLT`
- `cpu5.semihosting-Thumb_SVC`
- `cpu5.semihosting-cmd_line`
- `cpu5.semihosting-cwd`
- `cpu5.semihosting-enable`
- `cpu5.semihosting-heap_base`
- `cpu5.semihosting-heap_limit`
- `cpu5.semihosting-stack_base`
- `cpu5.semihosting-stack_limit`
- `cpu5.tcm.a.enable`
- `cpu5.tcm.a.size`
- `cpu5.tcm.a.wait`
- `cpu5.tcm.b.size`
- `cpu5.tcm.b.wait`
- `cpu5.tcm.c.size`

- `cpu5.tcm.c.wait`
- `cpu5.trace_special_hlt_imm16`
- `cpu5.vfp-dp-present`
- `cpu5.vfp-enable_at_reset`

About ARMCortexR52CT

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

- When you use the `semihosting-cmd_line` parameter to set the command line that is available to semihosting SVC calls, the value of `argv[0]` is set to the first command-line argument, not to the name of an image.
- The Cortex-R52 processor does not implement TrustZone technology, therefore the model does not support `S_*` or `NS_*` registers or exceptions.
- If flash memory is not enabled, to disable all routing to the flash port, set the `has_flash` parameter to `false`.


Differences between the model and the RTL

- The model does not implement redundant cores for Dual-Core Lock-Step operations.
- The model does not implement the Low Power Interface to wake up the target core on receiving a `wake_request` signal from the GIC distributor.
- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- The model does not support running Software Test Libraries (STLs).
- The `vfp-enable_at_reset` parameter is a model-specific behavior with no hardware equivalent.
- ECC and parity schemes are hardware-specific so are not supported.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

Debug accesses

For debug accesses to succeed, they must have permissions that are compliant with the permission value in the `IMP_SLAVEPCTLR` register.



Note

The reset value of `IMP_SLAVEPCTLR` is `0x1` which means privileged access only.

Iris and MTI instances for ARMCortexR52CT

This model has the following Iris instances:

Table 3-450: ARMCortexR52CT Iris instances

InstanceName	ComponentName
ARMCortexR52CT	Cluster_ARM_Cortex-R52

InstanceName	ComponentName
ARMCortexR52CT.AMU	PVBusLogger
ARMCortexR52CT.AMU.mapper	PVBusMapper
ARMCortexR52CT.DAP	PVBusLogger
ARMCortexR52CT.DAP.mapper	PVBusMapper
ARMCortexR52CT.MMAP	PVBusLogger
ARMCortexR52CT.MMAP.mapper	PVBusMapper
ARMCortexR52CT.RAS	PVBusLogger
ARMCortexR52CT.RAS.mapper	PVBusMapper
ARMCortexR52CT.acp_mapper	PVBusMapper
ARMCortexR52CT.cpu0	ARM_CortexR52
ARMCortexR52CT.cpu0.UTLB	TLB
ARMCortexR52CT.cpu0.dtlb	TLB
ARMCortexR52CT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR52CT.cpu0.l1dcache	PVCache
ARMCortexR52CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR52CT.cpu0.l1licache	PVCache
ARMCortexR52CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR52CT.ext_bus	PVBusLogger
ARMCortexR52CT.ext_bus.mapper	PVBusMapper
ARMCortexR52CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR52CT.gic_iri	GIC_IRI
ARMCortexR52CT.gic_iri.rd_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0_0_0	GICv3Redistributor
ARMCortexR52CT.gic_iri.rd_0_0_0_1	GICv3Redistributor
ARMCortexR52CT.gic_iri.rd_tl	GICv3Distributor
ARMCortexR52CT.global_debug_rom	debug_rom
ARMCortexR52CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-451: ARMCortexR52CT MTI instances**

InstanceName	ComponentName
ARMCortexR52CT.AMU	PVBusLogger
ARMCortexR52CT.AMU.mapper	PVBusMapper
ARMCortexR52CT.DAP	PVBusLogger
ARMCortexR52CT.DAP.mapper	PVBusMapper
ARMCortexR52CT.MMAP	PVBusLogger
ARMCortexR52CT.MMAP.mapper	PVBusMapper

InstanceName	ComponentName
ARMCortexR52CT.RAS	PVBusLogger
ARMCortexR52CT.RAS.mapper	PVBusMapper
ARMCortexR52CT.acp_mapper	PVBusMapper
ARMCortexR52CT.cpu0	ARM_CortexR52
ARMCortexR52CT.cpu0.UTLB	TLB
ARMCortexR52CT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR52CT.cpu0.l1dcache	PVCache
ARMCortexR52CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR52CT.cpu0.l1licache	PVCache
ARMCortexR52CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR52CT.ext_bus	PVBusLogger
ARMCortexR52CT.ext_bus.mapper	PVBusMapper
ARMCortexR52CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR52CT.gic_iri.rd_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0_0_0	GICv3Redistributor
ARMCortexR52CT.gic_iri.rd_0_0_0_1	GICv3Redistributor
ARMCortexR52CT.gic_iri.rd_tl	GICv3Distributor
ARMCortexR52CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexR52CT

**Table 3-452: Ports**

Name	Protocol	Type	Description
cfgdbgromaddr	Value_64	Slave	Debug ROM base address.
cfgdbgromaddrv	Signal	Slave	Debug ROM base address valid.
cfgendianess[4]	Signal	Slave	This signal if for EE bit initialisation.
cfgperiphbase	Value_64	Slave	This port sets the base address of private peripheral region.
cfgthumbexceptions[4]	Signal	Slave	This signal provides default exception handling state.
cfgvectable[4]	Value_64	Slave	Reset vector base address.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	Signal	Master	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	Signal	Slave	Signals the clearing of an external global exclusive monitor
clusterid	Value	Slave	Configure Aff2 and Aff1 fields of MPIDR. Aff1 = value[7:0], Aff2 = value[15:8]

Name	Protocol	Type	Description
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
commr[4]	Signal	Master	Receive portion of Data Transfer Register full.
commt[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
COREPACTIVEx1[4]	Signal	Master	These signals relate to core power down. Equivalent to COREPACTIVEx[1]
cpuhalt[4]	Signal	Slave	Raising this signal will put the core into halt mode.
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	There is no support for PChannel in CortexR52. These signals relate to core power down. Equivalent to COREPACTIVEx[0]
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
ext_slave_s	PVBus	Slave	External Slave port. Equivalent to AXIS port
extppi_in_0[9]	Signal	Slave	Core 0 external ppi signals.
extppi_in_1[9]	Signal	Slave	Core 1 external ppi signals.
extppi_in_2[9]	Signal	Slave	Core 2 external ppi signals.
extppi_in_3[9]	Signal	Slave	Core 3 external ppi signals.
flash_m[4]	PVBus	Master	Flash Port.
gdu_external_m	GICv3Comms	Master	GDU external messaging port.
hiden[4]	Signal	Slave	External debug interface.
hniden[4]	Signal	Slave	External debug interface.
llpp_m[4]	PVBus	Master	LLPP (Low-Latency Peripheral Port).
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_core_m[4]	PVBus	Master	The core will generate bus requests on this port. Equivalent to AXIM port
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
sei[4]	Signal	Slave	Per core virtual System Error physical pins.



Name	Protocol	Type	Description
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
topreset	Signal	Slave	This signal resets timer and interrupt controller.
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.
warmrstreq[4]	Signal	Master	Warm reset request from core.

## Parameters for ARMCortexR52CT

### CLUSTER\_ID

#### Type

int

#### Default value

0x0

#### Description

CLUSTER\_ID[15:8] equivalent to CFGMPIDRAFF2, CLUSTER\_ID[7:0] equivalent to CFGMPIDRAFF1.

### CMO\_broadcast\_when\_cache\_state\_modelling\_disabled

#### Type

int

#### Default value

0x1

#### Description

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

### DBGROMADDR

#### Type

int

**Default value**

0x0

**Description**

Equivalent to CFGDBGROMADDR.

**DBGROMADDRV****Type**

bool

**Default value**

0x0

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in cluster.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Equivalent to CFGPERIPHBASE.

**cluster\_utid****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGCLUSTERUTID.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGTHUMBEXCEPTIONS.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGVECTABLE.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has been built with NEON support.

**cpuX.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.flash.enable****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGFLASHEN.

**cpuX.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**cpuX.llpp.base****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGLLPPBASEADDR.

**cpuX.llpp.size****Type**

int

**Default value**

0x1000

**Description**

Equivalent to CFGLLPPSIZE.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.tcm.a.enable****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGTCMBOOT.

**cpuX.tcm.a.size****Type**

int

**Default value**

0x4000

**Description**

Sets the size of the ATCM(in bytes).

**cpuX.tcm.a.wait****Type**

int

**Default value**

0x0

**Description**

TCM Register A accesses wait states: 0-1 states.

**cpuX.tcm.b.size****Type**

int

**Default value**

0x4000



**Description**

Sets the size of the BTCM(in bytes).

**cpuX.tcm.b.wait****Type**

int

**Default value**

0x0

**Description**

TCM Register B accesses wait states: 0-1 states.

**cpuX.tcm.c.size****Type**

int

**Default value**

0x2000

**Description**

Sets the size of the CTCM(in bytes).

**cpuX.tcm.c.wait****Type**

int

**Default value**

0x0

**Description**

TCM Register C accesses wait states: 0-1 states.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-dp-present****Type**

bool

**Default value**

0x1

**Description**

Whether double-precision floating point feature is implemented (FEAT\_F64MM).

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**enable\_lock\_step****Type**

bool

**Default value**

0x0

**Description**

(equivalent to CFGSLSPPLIT).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**flash\_protection\_enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGFLASHPROTEN.

**has\_export\_m\_port****Type**

bool

**Default value**

0x1

**Description**

The interrupt distributor has an optional interrupt export port for routing interrupts to an external device.

**has\_flash****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGFLASHIMP.

**has\_flash\_protection****Type**

bool

**Default value**

0x1

**Description**

Equivalent to CFGFLASHPROTIMP.

**has\_llpp****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGLLPPIMP.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`memory.ext_slave_base`****Type**

int

**Default value**

0x0

**Description**

Equivalent to `CFGAXISTCMBASEADDR`.

**`memory.flash_base`****Type**

int

**Default value**

0x0

**Description**

Equivalent to `CFGFLASHBASEADDR`.

**num\_protection\_regions\_s1****Type**

int

**Default value**

0x18

**Description**

Number of v8-R protection regions.

**num\_protection\_regions\_s2****Type**

int

**Default value**

0x18

**Description**

Number of v8-R hyp protection regions.

**num\_spi****Type**

int

**Default value**

0x3c0

**Description**

Number of interrupts (SPI) into the internal GIC controller.

**ram\_protection\_enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGRAMPROTEN.

**reported\_fp\_revision****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

**reported\_patch\_level**

**Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number**

**Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

3.5.67 **ARMCortexR52PlusCT**

ARMCortexR52PlusCT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-453: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexR52PlusCT**

The model does not implement the following:

- Redundant cores for Dual Core Lock Step operation.
- Low Power Interface to wake the target core on receiving a wake\_request from the GIC Distributor.

## Debug accesses

For debug accesses to succeed, they must have permissions that are compliant with the permission value in the `IMP_SLAVEPCTLR` register.



Note

The reset value of `IMP_SLAVEPCTLR` is `0x1` which means privileged access only.

## Iris and MTI instances for ARMCortexR52PlusCT

This model has the following Iris instances:

**Table 3-454: ARMCortexR52PlusCT Iris instances**

InstanceName	ComponentName
ARMCortexR52PlusCT	Cluster_ARM_Cortex-R52Plus
ARMCortexR52PlusCT.AMU	PVBusLogger
ARMCortexR52PlusCT.AMU.mapper	PVBusMapper
ARMCortexR52PlusCT.DAP	PVBusLogger
ARMCortexR52PlusCT.DAP.mapper	PVBusMapper
ARMCortexR52PlusCT.MMAP	PVBusLogger
ARMCortexR52PlusCT.MMAP.mapper	PVBusMapper
ARMCortexR52PlusCT.RAS	PVBusLogger
ARMCortexR52PlusCT.RAS.mapper	PVBusMapper
ARMCortexR52PlusCT.acp_mapper	PVBusMapper
ARMCortexR52PlusCT.cpu0	ARM_Cortex-R52Plus
ARMCortexR52PlusCT.cpu0.UTLB	TLB
ARMCortexR52PlusCT.cpu0.dtlb	TLB
ARMCortexR52PlusCT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR52PlusCT.cpu0.l1dcache	PVCache
ARMCortexR52PlusCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR52PlusCT.cpu0.l1icache	PVCache
ARMCortexR52PlusCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR52PlusCT.ext_bus	PVBusLogger
ARMCortexR52PlusCT.ext_bus.mapper	PVBusMapper
ARMCortexR52PlusCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR52PlusCT.gic_iri	GIC_IRI
ARMCortexR52PlusCT.gic_iri.rd_0	GICv3RedistributorInternal
ARMCortexR52PlusCT.gic_iri.rd_0_0	GICv3RedistributorInternal
ARMCortexR52PlusCT.gic_iri.rd_0_0_0	GICv3RedistributorInternal
ARMCortexR52PlusCT.gic_iri.rd_0_0_0_0	GICv3Redistributor
ARMCortexR52PlusCT.gic_iri.rd_0_0_0_1	GICv3Redistributor



InstanceName	ComponentName
ARMCortexR52PlusCT.gic_iri.rd_tl	GICv3Distributor
ARMCortexR52PlusCT.global_debug_rom	debug_rom
ARMCortexR52PlusCT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-455: ARMCortexR52PlusCT MTI instances**

InstanceName	ComponentName
ARMCortexR52PlusCT.AMU	PVBusLogger
ARMCortexR52PlusCT.AMU.mapper	PVBusMapper
ARMCortexR52PlusCT.DAP	PVBusLogger
ARMCortexR52PlusCT.DAP.mapper	PVBusMapper
ARMCortexR52PlusCT.MMAP	PVBusLogger
ARMCortexR52PlusCT.MMAP.mapper	PVBusMapper
ARMCortexR52PlusCT.RAS	PVBusLogger
ARMCortexR52PlusCT.RAS.mapper	PVBusMapper
ARMCortexR52PlusCT.acp_mapper	PVBusMapper
ARMCortexR52PlusCT.cpu0	ARM_Cortex-R52Plus
ARMCortexR52PlusCT.cpu0.UTLB	TLB
ARMCortexR52PlusCT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR52PlusCT.cpu0.l1dcache	PVCache
ARMCortexR52PlusCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR52PlusCT.cpu0.l1icache	PVCache
ARMCortexR52PlusCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR52PlusCT.ext_bus	PVBusLogger
ARMCortexR52PlusCT.ext_bus.mapper	PVBusMapper
ARMCortexR52PlusCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR52PlusCT.gic_iri.rd_0	GICv3RedistributorInternal
ARMCortexR52PlusCT.gic_iri.rd_0_0	GICv3RedistributorInternal
ARMCortexR52PlusCT.gic_iri.rd_0_0_0	GICv3RedistributorInternal
ARMCortexR52PlusCT.gic_iri.rd_0_0_0_0	GICv3Redistributor
ARMCortexR52PlusCT.gic_iri.rd_0_0_0_1	GICv3Redistributor
ARMCortexR52PlusCT.gic_iri.rd_tl	GICv3Distributor
ARMCortexR52PlusCT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexR52PlusCT

**Table 3-456: Ports**

Name	Protocol	Type	Description
cfgdbgromaddr	Value_64	Slave	Debug ROM base address.
cfgdbgromaddrv	Signal	Slave	Debug ROM base address valid.

Name	Protocol	Type	Description
cfgendianess[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgperiphbase	Value_64	Slave	This port sets the base address of private peripheral region.
cfgthumbexceptions[4]	Signal	Slave	This signal provides default exception handling state.
cfgvectable[4]	Value_64	Slave	Reset vector base address.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	Configure Aff2 and Aff1 fields of MPIDR. Aff1 = value[7:0], Aff2 = value[15:8]
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
commrx[4]	Signal	Master	Receive portion of Data Transfer Register full.
commtx[4]	Signal	Master	Transmit portion of Data Transfer Register empty.
cpuhalt[4]	Signal	Slave	Raising this signal will put the core into halt mode.
cpuporeset[4]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
dbgack[4]	Signal	Master	External debug interface.
dbgen[4]	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	There is no support for PChannel in CortexR52Plus. These signals relate to core power down. Equivalent to COREPACTIVEx
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
edbgrq[4]	Signal	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
ext_slave_s	PVBus	Slave	External Slave port. Equivalent to AXIS port
extppi_in_0[9]	Signal	Slave	Core 0 external ppi signals.
extppi_in_1[9]	Signal	Slave	Core 1 external ppi signals.
extppi_in_2[9]	Signal	Slave	Core 2 external ppi signals.
extppi_in_3[9]	Signal	Slave	Core 3 external ppi signals.
flash_m[4]	PVBus	Master	Flash Port.
gdu_external_m	GICv3Comms	Master	GDU external messaging port.
hiden[4]	Signal	Slave	External debug interface.
hniden[4]	Signal	Slave	External debug interface.
llpp_m[4]	PVBus	Master	LLPP (Low-Latency Peripheral Port).
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden[4]	Signal	Slave	External debug interface.

Name	Protocol	Type	Description
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_core_m[4]	PVBus	Master	The core will generate bus requests on this port. Equivalent to AXIM port
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
sei[4]	Signal	Slave	Per core virtual System Error physical pins.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
standbywfe[4]	Signal	Master	This signal indicates if a core is in WFE state.
standbywfi[4]	Signal	Master	This signal indicates if a core is in WFI state.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
topreset	Signal	Slave	This signal resets timer and interrupt controller.
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.
warmrstreq[4]	Signal	Master	Warm reset request from core.

## Parameters for ARM Cortex R52 Plus CT

### CLUSTER\_ID

#### Type

int

#### Default value

0x0

#### Description

CLUSTER\_ID[15:8] equivalent to CFGMPIDRAFF2, CLUSTER\_ID[7:0] equivalent to CFGMPIDRAFF1.

### CMO\_broadcast\_when\_cache\_state\_modelling\_disabled

#### Type

int

#### Default value

0x1

#### Description

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not

broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGDBGROMADDR.

**DBGROMADDRV****Type**

bool

**Default value**

0x0

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in cluster.

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Equivalent to CFGPERIPHBASE.

**cluster\_utid****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGCLUSTERUTID.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGTHUMBEXCEPTIONS.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGVECTABLE.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has been built with NEON support.

**cpuX.dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.flash.enable****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGFLASHEN.

**cpuX.icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**cpuX.llpp.base****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGLLPPBASEADDR.

**cpuX.llpp.size****Type**

int

**Default value**

0x8000000

**Description**

Equivalent to CFGLLPPSIZE.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.



**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.tcm.a.enable****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGTCMBOOT.

**cpuX.tcm.a.size****Type**

int

**Default value**

0x4000

**Description**

Sets the size of the ATCM(in bytes).

**cpuX.tcm.a.wait****Type**

int

**Default value**

0x0

**Description**

TCM Register A accesses wait states: 0-1 states.

**cpuX.tcm.b.size****Type**

int

**Default value**

0x4000

**Description**

Sets the size of the BTCM(in bytes).

**cpuX.tcm.b.wait****Type**

int

**Default value**

0x0

**Description**

TCM Register B accesses wait states: 0-1 states.

**cpuX.tcm.c.size****Type**

int

**Default value**

0x2000

**Description**

Sets the size of the CTCM(in bytes).

**cpuX.tcm.c.wait****Type**

int

**Default value**

0x0

**Description**

TCM Register C accesses wait states: 0-1 states.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-dp-present****Type**

bool

**Default value**

0x1

**Description**

Whether double-precision floating point feature is implemented (FEAT\_F64MM).

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**enable\_lock\_step****Type**

bool

**Default value**

0x0

**Description**

(equivalent to CFGSLSPPLIT).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**flash\_protection\_enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGFLASHPROTEN.

**has\_export\_m\_port****Type**

bool

**Default value**

0x1

**Description**

The interrupt distributor has an optional interrupt export port for routing interrupts to an external device.

**has\_flash****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGFLASHIMP.

**has\_flash\_protection****Type**

bool

**Default value**

0x1

**Description**

Equivalent to CFGFLASHPROTIMP.

**has\_llpp****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGLLPPIMP.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**memory.ext\_slave\_base****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGAXISTCMBASEADDR.

**memory.flash\_base****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGFLASHBASEADDR.

**num\_protection\_regions\_s1****Type**

int

**Default value**

0x18

**Description**

Number of v8-R protection regions.

**num\_protection\_regions\_s2****Type**

int

**Default value**

0x18

**Description**

Number of v8-R hyp protection regions.

**num\_spi****Type**

int

**Default value**

0x3c0

**Description**

Number of interrupts (SPI) into the internal GIC controller.

**ram\_protection\_enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGRAMPROTEN.



**reported\_fp\_revision**

Type  
int

Default value  
0xffffffffffffffff

Description  
Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

**reported\_patch\_level**

Type  
int

Default value  
0xffffffffffffffff

Description  
Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number**

Type  
int

Default value  
0xffffffffffffffff

Description  
Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**3.5.68 ARMCortexR82AECT**

ARMCortexR82AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-457: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.
- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.

As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

## Iris and MTI instances for ARMCortexR82AECT

This model has the following Iris instances:

**Table 3-458: ARMCortexR82AECT Iris instances**

InstanceName	ComponentName
ARMCortexR82AECT	Cluster_ARM_Cortex-R82AE
ARMCortexR82AECT.AMU	PVBusLogger
ARMCortexR82AECT.AMU.mapper	PVBusMapper
ARMCortexR82AECT.DAP	PVBusLogger
ARMCortexR82AECT.DAP.mapper	PVBusMapper
ARMCortexR82AECT.DSU	DSU
ARMCortexR82AECT.DSU.PPU_cluster	PPUv1
ARMCortexR82AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexR82AECT.DSU.PPU_core0	PPUv1
ARMCortexR82AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexR82AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexR82AECT.MMAP	PVBusLogger
ARMCortexR82AECT.MMAP.mapper	PVBusMapper
ARMCortexR82AECT.RAS	PVBusLogger
ARMCortexR82AECT.RAS.mapper	PVBusMapper
ARMCortexR82AECT.cpu0	ARM_Cortex-R82AE
ARMCortexR82AECT.cpu0.UTLB	TLB
ARMCortexR82AECT.cpu0.debug_rom	debug_rom
ARMCortexR82AECT.cpu0.dtlb	TLB
ARMCortexR82AECT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR82AECT.cpu0.l1dcache	PVCache
ARMCortexR82AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR82AECT.cpu0.l1licache	PVCache

InstanceName	ComponentName
ARMCortexR82AECT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR82AECT.ext_bus	PVBusLogger
ARMCortexR82AECT.ext_bus.mapper	PVBusMapper
ARMCortexR82AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR82AECT.global_debug_rom	debug_rom
ARMCortexR82AECT.l2_cache	PVCache
ARMCortexR82AECT.l2_cache.upstream[0]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[10]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[11]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[12]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[13]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[14]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[15]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[16]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[1]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[2]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[3]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[4]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[5]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[6]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[7]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[8]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[9]	PVBusSlave
ARMCortexR82AECT.llram_atomic_filter	LLRAMAtomicOpFilter
ARMCortexR82AECT.llram_atomic_filter.atomic_bus_slave	PVBusSlave
ARMCortexR82AECT.llram_atomic_filter.filter	PVBusMapper
ARMCortexR82AECT.llram_atomic_filter.llram_exclusive_monitor	PVBusExclusiveMonitor
ARMCortexR82AECT.llram_atomic_filter.llram_exclusive_monitor.bus_mapper	PVBusMapper
ARMCortexR82AECT.llram_atomic_filter.pvbusmaster	PVBusMaster
ARMCortexR82AECT.llram_coherent_interconnect	PVCache
ARMCortexR82AECT.llram_coherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR82AECT.llram_coherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR82AECT.llram_coherent_interconnect.upstream[2]	PVBusSlave
ARMCortexR82AECT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-459: ARMCortexR82AECT MTI instances**

InstanceName	ComponentName
ARMCortexR82AECT.AMU	PVBusLogger

InstanceName	ComponentName
ARMCortexR82AECT.AMU.mapper	PVBusMapper
ARMCortexR82AECT.DAP	PVBusLogger
ARMCortexR82AECT.DAP.mapper	PVBusMapper
ARMCortexR82AECT.DSU	DSU
ARMCortexR82AECT.DSU.PPU_cluster	PPUv1
ARMCortexR82AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexR82AECT.DSU.PPU_core0	PPUv1
ARMCortexR82AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexR82AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexR82AECT.MMAP	PVBusLogger
ARMCortexR82AECT.MMAP.mapper	PVBusMapper
ARMCortexR82AECT.RAS	PVBusLogger
ARMCortexR82AECT.RAS.mapper	PVBusMapper
ARMCortexR82AECT.cpu0	ARM_Cortex-R82AE
ARMCortexR82AECT.cpu0.UTLB	TLB
ARMCortexR82AECT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR82AECT.cpu0.l1dcache	PVCache
ARMCortexR82AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR82AECT.cpu0.l1licache	PVCache
ARMCortexR82AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR82AECT.ext_bus	PVBusLogger
ARMCortexR82AECT.ext_bus.mapper	PVBusMapper
ARMCortexR82AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR82AECT.l2_cache	PVCache
ARMCortexR82AECT.l2_cache.upstream[0]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[10]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[11]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[12]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[13]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[14]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[15]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[16]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[1]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[2]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[3]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[4]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[5]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[6]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[7]	PVBusSlave

InstanceName	ComponentName
ARMCortexR82AECT.l2_cache.upstream[8]	PVBusSlave
ARMCortexR82AECT.l2_cache.upstream[9]	PVBusSlave
ARMCortexR82AECT.llram_atomic_filter	LLRAMAtomicOpFilter
ARMCortexR82AECT.llram_atomic_filter.atomic_bus_slave	PVBusSlave
ARMCortexR82AECT.llram_atomic_filter.filter	PVBusMapper
ARMCortexR82AECT.llram_atomic_filter.llram_exclusive_monitor	PVBusExclusiveMonitor
ARMCortexR82AECT.llram_atomic_filter.llram_exclusive_monitor.bus_mapper	PVBusMapper
ARMCortexR82AECT.llram_atomic_filter.pvbusmaster	PVBusMaster
ARMCortexR82AECT.llram_coherent_interconnect	PVCache
ARMCortexR82AECT.llram_coherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR82AECT.llram_coherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR82AECT.llram_coherent_interconnect.upstream[2]	PVBusSlave

## Ports for ARMCortexR82AECT

**Table 3-460: Ports**

Name	Protocol	Type	Description
acel_s	PVBus	Slave	External Slave port. Equivalent to AXIS port.
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
cache_validation_control	Value	Slave	This signal provides default exception handling state.
cfgendianess[8]	Signal	Slave	This signal if for EE bit initialisation
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPSIRQ[8]	Signal	Master	Timer signals to SOC
CNTHVSIRQ[8]	Signal	Master	Timer signals to SOC
CNTPSIRQ[8]	Signal	Master	Timer signals to SOC
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module
CNTVIRQ[8]	Signal	Master	Timer signals to SOC
commirq[8]	Signal	Master	Interrupt signal from debug communication channel.
core_pcsn_pchannel[8]	PChannel	Master	Core PCSM signals

Name	Protocol	Type	Description
coreerrirq[8]	Signal	Master	Core RAS error interrupt
corefaultirq[8]	Signal	Master	Core RAS fault interrupt
cp15sdisable[8]	Signal	Slave	This signal disables write access to some system control processor registers
cpuhalt[8]	Signal	Slave	Raising this signal will put the core into halt mode.
cti[8]	v8EmbeddedCrossTrigger_controlprotocol	Master	-
cti0extin[4]	Signal	Slave	CTI trace inputs for core 0.
cti0extout[4]	Signal	Master	CTI trace outputs for core 0.
cti1extin[4]	Signal	Slave	CTI trace inputs for core 1.
cti1extout[4]	Signal	Master	CTI trace outputs for core 1.
cti2extin[4]	Signal	Slave	CTI trace inputs for core 2.
cti2extout[4]	Signal	Master	CTI trace outputs for core 2.
cti3extin[4]	Signal	Slave	CTI trace inputs for core 3.
cti3extout[4]	Signal	Master	CTI trace outputs for core 3.
cti4extin[4]	Signal	Slave	CTI trace inputs for core 4.
cti4extout[4]	Signal	Master	CTI trace outputs for core 4.
cti5extin[4]	Signal	Slave	CTI trace inputs for core 5.
cti5extout[4]	Signal	Master	CTI trace outputs for core 5.
cti6extin[4]	Signal	Slave	CTI trace inputs for core 6.
cti6extout[4]	Signal	Master	CTI trace outputs for core 6.
cti7extin[4]	Signal	Slave	CTI trace inputs for core 7.
cti7extout[4]	Signal	Master	CTI trace outputs for core 7.
ctidbgirq[8]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	-
dbgnopwrdown[8]	Signal	Master	These signals relate to core power down.
dbgpwrdownack[8]	Signal	Master	Debug power down acknowledge.
dbgpwrdownreq[8]	Signal	Slave	Debug power down request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
external_trace_reset[8]	Signal	Slave	ETMv4 External Trace Reset signal.
fiq[8]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[8]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[8]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2reset	Signal	Slave	This signal resets timer and interrupt controller and l2cache

Name	Protocol	Type	Description
llpp_m[8]	PVBus	Master	LLPP (Low-Latency Peripheral Port).
llram_m	PVBus	Master	LLRAM Port
macp_s	PVBus	Slave	MACP slave interface
memorymapped_debug_s	PVBus	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region
pmuirq[8]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster irq signal
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wakeup request
ppu_core_irq[8]	Signal	Master	PPU core irq signal
ppu_core_wakerequest[8]	Signal	Slave	PPU core wakeup request
presetdbg[8]	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
reset	Signal	Slave	-
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
rvbar[8]	Value_64	Slave	Reset vector base address.
sei[8]	Signal	Slave	Per core System Error physical pins
spiden	Signal	Slave	Secure invasive debug enable.
spp_m	PVBus	Master	SPP (Shared Peripheral Port).
standbywfe[8]	Signal	Master	This signal indicates if a core is in WFE state
standbywfi[8]	Signal	Master	This signal indicates if a core is in WFI state
ticks[8]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trace_unit_reset[8]	Signal	Slave	ETMv4 Trace Unit Reset signal.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[8]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[8]	Signal	Slave	This signal drives the CPUs virtual fast-interrupt handling.
virq[8]	Signal	Slave	This signal drives the CPUs virtual interrupt handling.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).

## Parameters for ARMCortexR82AECT

### **BROADCASTATOMIC**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation to NC/DEV memory. The broadcastatomic signal will override this value if used.

### **BROADCASTATOMICL**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation to LLRAM memory. The broadcastatomicl signal will override this value and it will be functional for revision 2.

### **BROADCASTCACHEMAINT**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### **BROADCASTOUTER**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.



**CCSIDR-L1D\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L1I\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L2\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L3\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, allow L3 selection in CSSELR and present this value in CCSIDR (this is cosmetic and does not affect cache behaviour).

**CFGTFPEN\_pin\_reset****Type**

bool

**Default value**

0x0

**Description**

CFGTFPEN pin at reset.

**CHI****Type**

bool

**Default value**

0x0

**Description**

Selects the type of protocol the Main Manager(MM) interface implements. 0, MM port configured as AXI. 1, MM port configured as CHI.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

CLUSTER\_ID[15:8] equivalent to CFGMPIDRAFF3, CLUSTER\_ID[7:0] equivalent to CFGMPIDRAFF2.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x0

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in cluster.

**PA\_SIZE****Type**

int

**Default value**

0x28

**Description**

Physical address range supported (FEAT\_LPA).

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**VMSA\_supported****Type**

bool

**Default value**

0x1

**Description**

VMSA is supported at EL1.

**bus\_protection\_enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGBUSPROTEN.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x1

**Description**

Equivalent to PPU\_RST\_STATE. 0 = Cluster PPU and all core PPU's reset to OFF, 1 = Cluster PPU and all core PPU's reset to ON.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpu_mul`**

**Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`**

**Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CP15SDISABLE`**

**Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**`cpuX.RVBAR`**

**Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**`cpuX.TEINIT`**

**Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has been built with NEON support.

**cpuX.dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**cpuX.dtcn\_base****Type**

int

**Default value**

0x0

**Description**

Sets the 16K aligned base address of DTCM.

**cpuX.dtcn\_size****Type**

int

**Default value**

0x0

**Description**

Sets the size of DTCM (in bytes).

**cpuX.dtcn\_stretch\_clk****Type**

bool

**Default value**

0x0

**Description**

Whether DTCM clock stretched to occupy full cycle.

**cpuX.dtcn\_wait**

**Type**

int

**Default value**

0x0

**Description**

DTCM accesses wait states: 0-3 cycles.

**cpuX.enable\_trace\_special\_hlt\_imm16**

**Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.icache-size**

**Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**cpuX.itcm\_base**

**Type**

int

**Default value**

0x0

**Description**

Sets the 16K aligned base address of ITCM.

**cpuX.itcm\_size**

**Type**

int

**Default value**

0x0

**Description**

Sets the size of ITCM (in bytes).

**`cpuX.itcm_stretch_clk`****Type**

bool

**Default value**

0x0

**Description**

Whether ITCM clock stretched to occupy full cycle.

**`cpuX.itcm_wait`****Type**

int

**Default value**

0x0

**Description**

ITCM accesses wait states: 0-3 cycles.

**`cpuX.llpp.base`****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGLLPPBASEADDR.

**`cpuX.llpp.size`****Type**

int

**Default value**

0x8000000

**Description**

Equivalent to CFGLLPPSIZE.

**`cpuX.max_code_cache_mb`****Type**

int

**Default value**

0x100



**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.tcm.a.enable****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGITCMENm.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-dp-present****Type**

bool

**Default value**

0x1

**Description**

Whether double-precision floating point feature is implemented (FEAT\_F64MM).

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**enable\_lock\_step****Type**

bool

**Default value**

0x0

**Description**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**gicv3.BPR-min****Type**

int

**Default value**

0x2

**Description**

The minimum value for the GICC\_BPR register (non-secure version will be 1 + this value).

**gicv3.VBPR-min****Type**

int

**Default value**

0x2

**Description**

The minimum value for the GICV\_BPR register (non-secure version will be 1 + this value).

**has\_dense\_mem\_map****Type**

bool

**Default value**

0x0

**Description**

If true, the cluster follows the dense memory map else it implements the sparse memory map.

**has\_impdef\_transient\_fault\_protection****Type**

bool

**Default value**

0x1

**Description**

Support the Transient Fault Protection (TFP) flop parity errors through RAS registers (FEAT\_TFP).

**has\_llpp****Type**

bool

**Default value**

0x1

**Description**

Equivalent to CFGLLPPIMP.

**has\_pmc****Type**

bool

**Default value**

0x0

**Description**

Programmable MBIST controllers implemented.



**has\_spp****Type**

bool

**Default value**

0x1

**Description**

Equivalent to CFGSPPIMP.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`l2cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l2cache-size`****Type**

int

**Default value**

0x400000

**Description**

L2 Cache size in bytes.

**`l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**memory.ext\_slave\_base****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGACELSTCMBASEADDR.

**memory.has\_llram****Type**

bool

**Default value**

0x1

**Description**

Equivalent to CFGLLRAMIMP.

**memory.llram\_base****Type**

int

**Default value**

0x20000000

**Description**

Equivalent to CFGLLRAMBASEADDR.

**memory.llram\_enable\_at\_reset****Type**

bool

**Default value**

0x1

**Description**

Equivalent to CFGLLRAMEN.

**memory.llram\_shared****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGLLRAMSHARED and it is only functional for revision 2.

**memory.llram\_size****Type**

int

**Default value**

0x10000000

**Description**

Size of the LLRAM.

**num\_protection\_regions\_s1****Type**

int

**Default value**

0x10

**Description**

Number of v8-R protection regions.

**num\_protection\_regions\_s2****Type**

int

**Default value**

0x10

**Description**

Number of v8-R hyp protection regions.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**ram\_protection\_enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGRAMPROTEN.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**spp.base****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGSPPBASEADDR.

**spp.size****Type**

int

**Default value**

0x8000000

**Description**

Sets the size of SPP(in bytes).

**stage12\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

If VMSA is supported at stage1, number of stage1+2 tlb entries. If instruction\_tlb\_size !=0, this is treated as dtlb size.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

## 3.5.69 ARM CortexR82CT

ARM CortexR82CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:



**Table 3-461: IP revisions support**

Revision	Quality level
r0p0	Preliminary support
r1p1	Preliminary support
r2p1	Preliminary support
r3p1	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About ARMCortexR82CT

To simulate the r1p0 model, use the following parameters:

- `revision_number=1`
- `VMSA_supported=1`

### Limitations

Dense memory map support has been added for the Utility bus only.

### Iris and MTI instances for ARMCortexR82CT

This model has the following Iris instances:

**Table 3-462: ARMCortexR82CT Iris instances**

InstanceName	ComponentName
ARMCortexR82CT	Cluster_ARM_Cortex-R82
ARMCortexR82CT.AMU	PVBusLogger
ARMCortexR82CT.AMU.mapper	PVBusMapper
ARMCortexR82CT.DAP	PVBusLogger
ARMCortexR82CT.DAP.mapper	PVBusMapper
ARMCortexR82CT.DSU	DSU
ARMCortexR82CT.DSU.PPU_cluster	PPUv1
ARMCortexR82CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexR82CT.DSU.PPU_core0	PPUv1
ARMCortexR82CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexR82CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexR82CT.MMAP	PVBusLogger
ARMCortexR82CT.MMAP.mapper	PVBusMapper
ARMCortexR82CT.RAS	PVBusLogger
ARMCortexR82CT.RAS.mapper	PVBusMapper
ARMCortexR82CT.cpu0	ARM_Cortex-R82
ARMCortexR82CT.cpu0.UTLB	TLB
ARMCortexR82CT.cpu0.debug_rom	debug_rom
ARMCortexR82CT.cpu0.dtlb	TLB

InstanceName	ComponentName
ARMCortexR82CT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR82CT.cpu0.l1dcache	PVCache
ARMCortexR82CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR82CT.cpu0.l1icache	PVCache
ARMCortexR82CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR82CT.ext_bus	PVBusLogger
ARMCortexR82CT.ext_bus.mapper	PVBusMapper
ARMCortexR82CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR82CT.global_debug_rom	debug_rom
ARMCortexR82CT.l2_cache	PVCache
ARMCortexR82CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[3]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexR82CT.llram_atomic_filter	LLRAMAtomicOpFilter
ARMCortexR82CT.llram_atomic_filter.atomic_bus_slave	PVBusSlave
ARMCortexR82CT.llram_atomic_filter.filter	PVBusMapper
ARMCortexR82CT.llram_atomic_filter.llram_exclusive_monitor	PVBusExclusiveMonitor
ARMCortexR82CT.llram_atomic_filter.llram_exclusive_monitor.bus_mapper	PVBusMapper
ARMCortexR82CT.llram_atomic_filter.pvbusmaster	PVBusMaster
ARMCortexR82CT.llram_coherent_interconnect	PVCache
ARMCortexR82CT.llram_coherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR82CT.llram_coherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR82CT.llram_coherent_interconnect.upstream[2]	PVBusSlave
ARMCortexR82CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-463: ARMCortexR82CT MTI instances**

InstanceName	ComponentName
ARMCortexR82CT.AMU	PVBusLogger
ARMCortexR82CT.AMU.mapper	PVBusMapper
ARMCortexR82CT.DAP	PVBusLogger
ARMCortexR82CT.DAP.mapper	PVBusMapper
ARMCortexR82CT.DSU	DSU
ARMCortexR82CT.DSU.PPU_cluster	PPUv1
ARMCortexR82CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexR82CT.DSU.PPU_core0	PPUv1
ARMCortexR82CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexR82CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexR82CT.MMAP	PVBusLogger
ARMCortexR82CT.MMAP.mapper	PVBusMapper
ARMCortexR82CT.RAS	PVBusLogger
ARMCortexR82CT.RAS.mapper	PVBusMapper
ARMCortexR82CT.cpu0	ARM_Cortex-R82
ARMCortexR82CT.cpu0.UTLB	TLB
ARMCortexR82CT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR82CT.cpu0.l1dcache	PVCache
ARMCortexR82CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR82CT.cpu0.l1icache	PVCache
ARMCortexR82CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR82CT.ext_bus	PVBusLogger
ARMCortexR82CT.ext_bus.mapper	PVBusMapper
ARMCortexR82CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR82CT.l2_cache	PVCache
ARMCortexR82CT.l2_cache.upstream[0]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[10]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[11]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[12]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[13]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[14]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[15]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[16]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[1]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[2]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[3]	PVBusSlave

InstanceName	ComponentName
ARMCortexR82CT.l2_cache.upstream[4]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[5]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[6]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[7]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[8]	PVBusSlave
ARMCortexR82CT.l2_cache.upstream[9]	PVBusSlave
ARMCortexR82CT.llram_atomic_filter	LLRAMAtomicOpFilter
ARMCortexR82CT.llram_atomic_filter.atomic_bus_slave	PVBusSlave
ARMCortexR82CT.llram_atomic_filter.filter	PVBusMapper
ARMCortexR82CT.llram_atomic_filter.llram_exclusive_monitor	PVBusExclusiveMonitor
ARMCortexR82CT.llram_atomic_filter.llram_exclusive_monitor.bus_mapper	PVBusMapper
ARMCortexR82CT.llram_atomic_filter.pvbusmaster	PVBusMaster
ARMCortexR82CT.llram_coherent_interconnect	PVCache
ARMCortexR82CT.llram_coherent_interconnect.upstream[0]	PVBusSlave
ARMCortexR82CT.llram_coherent_interconnect.upstream[1]	PVBusSlave
ARMCortexR82CT.llram_coherent_interconnect.upstream[2]	PVBusSlave

## Ports for ARMCortexR82CT

**Table 3-464: Ports**

Name	Protocol	Type	Description
acel_s	PVBus	Slave	External Slave port. Equivalent to AXIS port.
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastatomicl	Signal	Slave	BROADCASTATOMIC pin for LLRAM
broadcastcachemaint	Signal	Slave	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastouter	Signal	Slave	Enable broadcasting of Outer Shareable transactions.
cache_validation_control	Value	Slave	This signal provides default exception handling state.
cfgendianess[8]	Signal	Slave	This signal if for EE bit initialisation
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
clusterid	Value	Slave	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPSIRQ[8]	Signal	Master	Timer signals to SOC
CNTHVSIRQ[8]	Signal	Master	Timer signals to SOC
CNTPSIRQ[8]	Signal	Master	Timer signals to SOC

Name	Protocol	Type	Description
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module
CNTVIRQ[8]	Signal	Master	Timer signals to SOC
commirq[8]	Signal	Master	Interrupt signal from debug communication channel.
core_pcs_m_pchannel[8]	PChannel	Master	Core PCSM signals
coreerrirq[8]	Signal	Master	Core RAS error interrupt
corefaultirq[8]	Signal	Master	Core RAS fault interrupt
cp15sdisable[8]	Signal	Slave	This signal disables write access to some system control processor registers
cpuhalt[8]	Signal	Slave	Raising this signal will put the core into halt mode.
cti[8]	v8EmbeddedCrossTrigger_controlprotocol	Master	-
cti0extin[4]	Signal	Slave	CTI trace inputs for core 0.
cti0extout[4]	Signal	Master	CTI trace outputs for core 0.
cti1extin[4]	Signal	Slave	CTI trace inputs for core 1.
cti1extout[4]	Signal	Master	CTI trace outputs for core 1.
cti2extin[4]	Signal	Slave	CTI trace inputs for core 2.
cti2extout[4]	Signal	Master	CTI trace outputs for core 2.
cti3extin[4]	Signal	Slave	CTI trace inputs for core 3.
cti3extout[4]	Signal	Master	CTI trace outputs for core 3.
cti4extin[4]	Signal	Slave	CTI trace inputs for core 4.
cti4extout[4]	Signal	Master	CTI trace outputs for core 4.
cti5extin[4]	Signal	Slave	CTI trace inputs for core 5.
cti5extout[4]	Signal	Master	CTI trace outputs for core 5.
cti6extin[4]	Signal	Slave	CTI trace inputs for core 6.
cti6extout[4]	Signal	Master	CTI trace outputs for core 6.
cti7extin[4]	Signal	Slave	CTI trace inputs for core 7.
cti7extout[4]	Signal	Master	CTI trace outputs for core 7.
ctidbgirq[8]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	-
dbgnopwrdown[8]	Signal	Master	These signals relate to core power down.
dbgpwrdownack[8]	Signal	Master	Debug power down acknowledge.
dbgpwrdownreq[8]	Signal	Slave	Debug power down request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
external_trace_reset[8]	Signal	Slave	ETMv4 External Trace Reset signal.
fiq[8]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.

Name	Protocol	Type	Description
gicv3_redistributor_s[8]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[8]	Signal	Slave	This signal drives the CPUs interrupt handling.
l2reset	Signal	Slave	This signal resets timer and interrupt controller and l2cache
llpp_m[8]	PVBus	Master	LLPP (Low-Latency Peripheral Port).
llram_m	PVBus	Master	LLRAM Port
macp_s	PVBus	Slave	MACP slave interface
memorymapped_debug_s	PVBus	Slave	External debug interface.
periphbase	Value_64	Slave	This port sets the base address of private peripheral region
pmuirq[8]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster irq signal
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wakeup request
ppu_core_irq[8]	Signal	Master	PPU core irq signal
ppu_core_wakerequest[8]	Signal	Slave	PPU core wakeup request
presetdbg[8]	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
reset	Signal	Slave	-
romaddr	Value_64	Slave	Debug ROM base address.
romaddrv	Signal	Slave	Debug ROM base address valid.
rvbar[8]	Value_64	Slave	Reset vector base address.
sei[8]	Signal	Slave	Per core System Error physical pins
spiden	Signal	Slave	Secure invasive debug enable.
spp_m	PVBus	Master	SPP (Shared Peripheral Port).
standbywfe[8]	Signal	Master	This signal indicates if a core is in WFE state
standbywfi[8]	Signal	Master	This signal indicates if a core is in WFI state
ticks[8]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trace_unit_reset[8]	Signal	Slave	ETMv4 Trace Unit Reset signal.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[8]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[8]	Signal	Slave	This signal drives the CPUs virtual fast-interrupt handling.

Name	Protocol	Type	Description
virq[8]	Signal	Slave	This signal drives the CPUs virtual interrupt handling.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).

## Parameters for ARM CortexR82CT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation to NC/DEV memory. The broadcastatomic signal will override this value if used.

### BROADCASTATOMICL

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation to LLRAM memory. The broadcastatomicl signal will override this value and it will be functional for revision 2.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTOUTER

#### Type

bool

#### Default value

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**CCSIDR-L1D\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L1I\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L2\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

**CCSIDR-L3\_override****Type**

int

**Default value**

0x0

**Description**

If nonzero, allow L3 selection in CSSELR and present this value in CCSIDR (this is cosmetic and does not affect cache behaviour).



**CHI****Type**

bool

**Default value**

0x0

**Description**

Selects the type of protocol the Main Manager(MM) interface implements. 0, MM port configured as AXI. 1, MM port configured as CHI.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

CLUSTER\_ID[15:8] equivalent to CFGMPIDRAFF3, CLUSTER\_ID[7:0] equivalent to CFGMPIDRAFF2.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**DBGROMADDR****Type**

int

**Default value**

0x0

**Description**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

**DBGROMADDRV****Type**

bool

**Default value**

0x0

**Description**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

**GICDISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores in cluster.

**PA\_SIZE****Type**

int

**Default value**

0x28

**Description**

Physical address range supported (FEAT\_LPA).

**PERIPHBASE****Type**

int

**Default value**

0x13080000

**Description**

Base address of peripheral memory space.

**VMSA\_supported****Type**

bool

**Default value**

0x0

**Description**

VMSA is supported at EL1.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x1

**Description**

Equivalent to PPU\_RST\_STATE. 0 = Cluster PPU and all core PPUs reset to OFF, 1 = Cluster PPU and all core PPUs reset to ON.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CP15SDISABLE****Type**

bool

**Default value**

0x0

**Description**

Initialize to disable access to some CP15 registers.

**cpuX.RVBAR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.TEINIT****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.ase-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has been built with NEON support.

**cpuX.dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**cpuX.dtcn\_base****Type**

int

**Default value**

0x0

**Description**

Sets the 16K aligned base address of DTCM.

**cpuX.dtcn\_size****Type**

int

**Default value**

0x0

**Description**

Sets the size of DTCM (in bytes).

**cpuX.dtcn\_stretch\_clk****Type**

bool

**Default value**

0x0

**Description**

Whether DTCM clock stretched to occupy full cycle.

**cpuX.dtcn\_wait****Type**

int

**Default value**

0x0

**Description**

DTCM accesses wait states: 0-3 cycles.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**cpuX.itcm\_base****Type**

int

**Default value**

0x0

**Description**

Sets the 16K aligned base address of ITCM.

**cpuX.itcm\_size****Type**

int

**Default value**

0x0

**Description**

Sets the size of ITCM (in bytes).

**cpuX.itcm\_stretch\_clk****Type**

bool

**Default value**

0x0

**Description**

Whether ITCM clock stretched to occupy full cycle.

**cpuX.itcm\_wait****Type**

int

**Default value**

0x0

**Description**

ITCM accesses wait states: 0-3 cycles.

**cpuX.llpp.base****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGLLPPBASEADDR.

**cpuX.llpp.size****Type**

int

**Default value**

0x8000000

**Description**

Equivalent to CFGLLPPSIZE.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.



**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.tcm.a.enable****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGITCMENm.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-dp-present****Type**

bool

**Default value**

0x1

**Description**

Whether double-precision floating point feature is implemented (FEAT\_F64MM).

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**enable\_lock\_step****Type**

bool

**Default value**

0x0

**Description**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**gicv3.BPR-min****Type**

int

**Default value**

0x2

**Description**

The minimum value for the GICC\_BPR register (non-secure version will be 1 + this value).

**gicv3.VBPR-min****Type**

int

**Default value**

0x2

**Description**

The minimum value for the GICV\_BPR register (non-secure version will be 1 + this value).

**has\_dense\_mem\_map****Type**

bool

**Default value**

0x0

**Description**

If true, the cluster follows the dense memory map else it implements the sparse memory map.

**has\_llpp****Type**

bool

**Default value**

0x1

**Description**

Equivalent to CFGLLPPIMP.

**has\_spp****Type**

bool

**Default value**

0x1

**Description**

Equivalent to CFGSPPIMP.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.



**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l2cache-size`****Type**

int

**Default value**

0x400000

**Description**

L2 Cache size in bytes.

**`l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**`l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**`l2cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**`memory.ext_slave_base`****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGACELSTCMBASEADDR.

**`memory.has_llram`****Type**

bool

**Default value**

0x1

**Description**

Equivalent to CFGLLRAMIMP.

**memory.llram\_base****Type**

int

**Default value**

0x20000000

**Description**

Equivalent to CFGLLRAMBASEADDR.

**memory.llram\_enable\_at\_reset****Type**

bool

**Default value**

0x1

**Description**

Equivalent to CFGLLRAMEN.

**memory.llram\_shared****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGLLRAMSHARED and it is only functional for revision 2.

**memory.llram\_size****Type**

int

**Default value**

0x10000000

**Description**

Size of the LLRAM.

**num\_protection\_regions\_s1****Type**

int

**Default value**

0x10

**Description**

Number of v8-R protection regions.

**num\_protection\_regions\_s2****Type**

int

**Default value**

0x10

**Description**

Number of v8-R hyp protection regions.

**patch\_level****Type**

int

**Default value**

0x1

**Description**

Patch level of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Revision field in MIDR/MIDR\_EL1. Corresponds to the patch number Y in rXpY.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**ram\_protection\_enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Equivalent to CFGRAMPROTEN.

**revision\_number****Type**

int

**Default value**

0x1

**Description**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

**spp.base****Type**

int

**Default value**

0x0

**Description**

Equivalent to CFGSPPBASEADDR.

**spp.size****Type**

int

**Default value**

0x8000000

**Description**

Sets the size of SPP(in bytes).

**stage12\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

If VMSA is supported at stage1, number of stage1+2 tlb entries. If instruction\_tlb\_size !=0, this is treated as dtlb size.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.70 ARMCortexX1CCT**

ARMCortexX1CCT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-465: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexX1CCT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers’ view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).

- Latency configuration.
- Snoop filtering.
- Cache stashing capability.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

**Note**

The `cfgsdisable` signal will be removed in a future release.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARMCortexX1CCT

This model has the following Iris instances:

**Table 3-466: ARMCortexX1CCT Iris instances**

InstanceName	ComponentName
ARMCortexX1CCT	Cluster_ARM_Cortex-X1C
ARMCortexX1CCT.AMU	PVBusLogger
ARMCortexX1CCT.AMU.mapper	PVBusMapper
ARMCortexX1CCT.DAP	PVBusLogger
ARMCortexX1CCT.DAP.mapper	PVBusMapper
ARMCortexX1CCT.DSU	DSU
ARMCortexX1CCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX1CCT.DSU.mpam_busslave	PVBusSlave
ARMCortexX1CCT.DSU.shared_cache	PVCache
ARMCortexX1CCT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX1CCT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX1CCT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX1CCT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX1CCT.MMAP	PVBusLogger
ARMCortexX1CCT.MMAP.mapper	PVBusMapper
ARMCortexX1CCT.RAS	PVBusLogger
ARMCortexX1CCT.RAS.mapper	PVBusMapper
ARMCortexX1CCT.cpu0	ARM_Cortex-X1C
ARMCortexX1CCT.cpu0.UTLB	TLB
ARMCortexX1CCT.cpu0.debug_rom	debug_rom



InstanceName	ComponentName
ARMCortexX1CCT.cpu0.dtlb	TLB
ARMCortexX1CCT.cpu0.l1dcache	PVCache
ARMCortexX1CCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX1CCT.cpu0.l1icache	PVCache
ARMCortexX1CCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX1CCT.cpu0.l2cache	PVCache
ARMCortexX1CCT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexX1CCT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX1CCT.ext_bus	PVBusLogger
ARMCortexX1CCT.ext_bus.mapper	PVBusMapper
ARMCortexX1CCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX1CCT.global_debug_rom	debug_rom
ARMCortexX1CCT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-467: ARMCortexX1CCT MTI instances**

InstanceName	ComponentName
ARMCortexX1CCT.AMU	PVBusLogger
ARMCortexX1CCT.AMU.mapper	PVBusMapper
ARMCortexX1CCT.DAP	PVBusLogger
ARMCortexX1CCT.DAP.mapper	PVBusMapper
ARMCortexX1CCT.DSU	DSU
ARMCortexX1CCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX1CCT.DSU.mpam_busslave	PVBusSlave
ARMCortexX1CCT.DSU.shared_cache	PVCache
ARMCortexX1CCT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX1CCT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX1CCT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX1CCT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX1CCT.MMAP	PVBusLogger
ARMCortexX1CCT.MMAP.mapper	PVBusMapper
ARMCortexX1CCT.RAS	PVBusLogger
ARMCortexX1CCT.RAS.mapper	PVBusMapper
ARMCortexX1CCT.cpu0	ARM_Cortex-X1C
ARMCortexX1CCT.cpu0.UTLB	TLB
ARMCortexX1CCT.cpu0.l1dcache	PVCache
ARMCortexX1CCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX1CCT.cpu0.l1icache	PVCache
ARMCortexX1CCT.cpu0.l1icache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexX1CCT.cpu0.l2cache	PVCache
ARMCortexX1CCT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexX1CCT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX1CCT.ext_bus	PVBusLogger
ARMCortexX1CCT.ext_bus.mapper	PVBusMapper
ARMCortexX1CCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexX1CCT

**Table 3-468: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[8]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[8]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[8]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[8]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[8]	Signal	Master	Timer signals to SOC.

Name	Protocol	Type	Description
CNTPSIRQ[8]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[8]	Signal	Master	Timer signals to SOC.
commirq[8]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[8]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[8]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[8]	Signal	Slave	Disable cryptography extensions after reset.
cti[8]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[8]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[8]	Signal	Master	No power-down request.
dbgprupreq[8]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[8]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[8]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[8]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[8]	PChannel	Slave	PChannels for cores
pmbirq[8]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[8]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[8]	Signal	Slave	Per core RAM Error Interrupt.
reset[8]	Signal	Slave	Raising this signal will put the core into reset mode.

Name	Protocol	Type	Description
rvbaraddr[8]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[8]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[8]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[8]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[8]	Signal	Slave	Virtualised FIQ.
vinithi[8]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[8]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[8]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM CortexX1CCT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are

broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-



read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

### **cpuX.l2cache-read\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

### **cpuX.l2cache-size**

#### **Type**

int

#### **Default value**

0x80000

#### **Description**

L2 Cache size in bytes.

### **cpuX.l2cache-snoop\_data\_transfer\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

### **cpuX.l2cache-snoop\_issue\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**`cpuX.semihosting-stack_limit`****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**`cpuX.trace_special_hlt_imm16`****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

**`cpuX.vfp-enable_at_reset`****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**`dcache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`dcache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`default_opmode`****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**`diagnostics`****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**`enable_simulation_performance_optimizations`****Type**

bool

**Default value**

0x1



**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_coherent\_icache****Type**

bool

**Default value**

0x0

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x100000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

**`periph_address_start`****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

**`ptw_latency`****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**`tlb_latency`****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**`tlbi_stall_enabled`****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.



**treat-dcache-cmos-to-pou-as-nop**

**Type**  
int

**Default value**  
0x0

**Description**  
Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**treat\_PAC\_as\_NOP**

**Type**  
bool

**Default value**  
0x0

**Description**  
Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency**

**Type**  
int

**Default value**  
0x0

**Description**  
Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.71 ARMCortexX1CT

ARMCortexX1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-469: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About ARMCortexX1CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGPWRDUP and DBGIRSTREQ are not implemented, but DBGPWRUPREQ and DBGNOPWRDWN are implemented.
- Cache stashing capability.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where n identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMCortexX1CT

This model has the following Iris instances:

**Table 3-470: ARMCortexX1CT Iris instances**

InstanceName	ComponentName
ARMCortexX1CT	Cluster_ARM_Cortex-X1
ARMCortexX1CT.AMU	PVBusLogger
ARMCortexX1CT.AMU.mapper	PVBusMapper
ARMCortexX1CT.DAP	PVBusLogger
ARMCortexX1CT.DAP.mapper	PVBusMapper
ARMCortexX1CT.DSU	DSU
ARMCortexX1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX1CT.DSU.shared_cache	PVCache
ARMCortexX1CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX1CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX1CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX1CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX1CT.MMAP	PVBusLogger
ARMCortexX1CT.MMAP.mapper	PVBusMapper
ARMCortexX1CT.RAS	PVBusLogger
ARMCortexX1CT.RAS.mapper	PVBusMapper
ARMCortexX1CT.cpu0	ARM_Cortex-X1
ARMCortexX1CT.cpu0.UTLB	TLB
ARMCortexX1CT.cpu0.debug_rom	debug_rom
ARMCortexX1CT.cpu0.dtlb	TLB
ARMCortexX1CT.cpu0.l1dcache	PVCache
ARMCortexX1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX1CT.cpu0.l1icache	PVCache
ARMCortexX1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX1CT.cpu0.l2cache	PVCache
ARMCortexX1CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexX1CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX1CT.ext_bus	PVBusLogger
ARMCortexX1CT.ext_bus.mapper	PVBusMapper
ARMCortexX1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX1CT.global_debug_rom	debug_rom
ARMCortexX1CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-471: ARMCortexX1CT MTI instances**

InstanceName	ComponentName
ARMCortexX1CT.AMU	PVBusLogger

InstanceName	ComponentName
ARMCortexX1CT.AMU.mapper	PVBusMapper
ARMCortexX1CT.DAP	PVBusLogger
ARMCortexX1CT.DAP.mapper	PVBusMapper
ARMCortexX1CT.DSU	DSU
ARMCortexX1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX1CT.DSU.shared_cache	PVCache
ARMCortexX1CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX1CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX1CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX1CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX1CT.MMAP	PVBusLogger
ARMCortexX1CT.MMAP.mapper	PVBusMapper
ARMCortexX1CT.RAS	PVBusLogger
ARMCortexX1CT.RAS.mapper	PVBusMapper
ARMCortexX1CT.cpu0	ARM_Cortex-X1
ARMCortexX1CT.cpu0.UTLB	TLB
ARMCortexX1CT.cpu0.l1dcache	PVCache
ARMCortexX1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX1CT.cpu0.l1icache	PVCache
ARMCortexX1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX1CT.cpu0.l2cache	PVCache
ARMCortexX1CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexX1CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX1CT.ext_bus	PVBusLogger
ARMCortexX1CT.ext_bus.mapper	PVBusMapper
ARMCortexX1CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

## Ports for ARMCortexX1CT

**Table 3-472: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.

Name	Protocol	Type	Description
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[4]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[4]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.

Name	Protocol	Type	Description
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.

Name	Protocol	Type	Description
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM CortexX1CT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTOUTER

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

### BROADCASTPERSIST

#### Type

bool

#### Default value

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).



**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`cpuX.l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**`cpuX.semihosting-ARM_SVC`****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**`cpuX.semihosting-T32_HLT`****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**`cpuX.semihosting-Thumb_SVC`****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**`cpuX.semihosting-cmd_line`****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**`cpuX.semihosting-cwd`****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**`cpuX.trace_special_hlt_imm16`****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

**`cpuX.vfp-enable_at_reset`****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**`dcache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`dcache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.



**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_coherent\_icache****Type**

bool

**Default value**

0x0

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l3cache-read\_latency****Type**

int

**Default value**

0x0



**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l3cache-size`****Type**

int

**Default value**

0x100000

**Description**

L3 Cache size in bytes.

**`l3cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l3cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l3cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

### **l3cache-write\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

### **pchannel\_treat\_simreset\_as\_poreset**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Register core as ON state to cluster with simulation reset.

### **periph\_address\_end**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

### **periph\_address\_start**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.72 **ARMCortexX2CT**

ARMCortexX2CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-473: IP revisions support**

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexX2CT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

Support for the following features is planned for a future release:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, and nPMBIRQ signals.
- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (TLM2 extensions from Arm) bus protocols.

The following features will not be implemented:

- 256-bit wide output transactions.
- Error correction/detection features.
- Self-test features (MBIST).
- Snoop filtering.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMCortexX2CT

This model has the following Iris instances:

**Table 3-474: ARMCortexX2CT Iris instances**

InstanceName	ComponentName
ARMCortexX2CT	Cluster_ARM_Cortex-X2
ARMCortexX2CT.AMU	PVBusLogger
ARMCortexX2CT.AMU.mapper	PVBusMapper
ARMCortexX2CT.DAP	PVBusLogger
ARMCortexX2CT.DAP.mapper	PVBusMapper
ARMCortexX2CT.DSU	DSU
ARMCortexX2CT.DSU.PPU_cluster	PPUv1
ARMCortexX2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX2CT.DSU.PPU_core0	PPUv1
ARMCortexX2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX2CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX2CT.DSU.shared_cache	PVCache
ARMCortexX2CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX2CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX2CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX2CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX2CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX2CT.MMAP	PVBusLogger
ARMCortexX2CT.MMAP.mapper	PVBusMapper
ARMCortexX2CT.RAS	PVBusLogger
ARMCortexX2CT.RAS.mapper	PVBusMapper
ARMCortexX2CT.cpu0	ARM_Cortex-X2
ARMCortexX2CT.cpu0.UTLB	TLB
ARMCortexX2CT.cpu0.debug_rom	debug_rom
ARMCortexX2CT.cpu0.dtlb	TLB
ARMCortexX2CT.cpu0.l1dcache	PVCache
ARMCortexX2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX2CT.cpu0.l1licache	PVCache
ARMCortexX2CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexX2CT.cpu0.l2cache	PVCache
ARMCortexX2CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexX2CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX2CT.ext_bus	PVBusLogger
ARMCortexX2CT.ext_bus.mapper	PVBusMapper
ARMCortexX2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

InstanceName	ComponentName
ARMCortexX2CT.global_debug_rom	debug_rom
ARMCortexX2CT.secondary_debug_rom	debug_rom
ARMCortexX2CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-475: ARMCortexX2CT MTI instances**

InstanceName	ComponentName
ARMCortexX2CT.AMU	PVBusLogger
ARMCortexX2CT.AMU.mapper	PVBusMapper
ARMCortexX2CT.DAP	PVBusLogger
ARMCortexX2CT.DAP.mapper	PVBusMapper
ARMCortexX2CT.DSU	DSU
ARMCortexX2CT.DSU.PPU_cluster	PPUv1
ARMCortexX2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX2CT.DSU.PPU_core0	PPUv1
ARMCortexX2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX2CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX2CT.DSU.shared_cache	PVCache
ARMCortexX2CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX2CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX2CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX2CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX2CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX2CT.MMAP	PVBusLogger
ARMCortexX2CT.MMAP.mapper	PVBusMapper
ARMCortexX2CT.RAS	PVBusLogger
ARMCortexX2CT.RAS.mapper	PVBusMapper
ARMCortexX2CT.cpu0	ARM_Cortex-X2
ARMCortexX2CT.cpu0.UTLB	TLB
ARMCortexX2CT.cpu0.l1dcache	PVCache
ARMCortexX2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX2CT.cpu0.l1licache	PVCache
ARMCortexX2CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexX2CT.cpu0.l2cache	PVCache
ARMCortexX2CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexX2CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX2CT.ext_bus	PVBusLogger
ARMCortexX2CT.ext_bus.mapper	PVBusMapper
ARMCortexX2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexX2CT

Table 3-476: Ports

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[12]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[12]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[12]	Signal	Master	Timer signals to SOC
CNTHVIRQ[12]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[12]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[12]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[12]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[12]	Signal	Master	Timer signals to SOC.
commirq[12]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[12]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[12]	PChannel	Master	Core PCSM signals
core_powerdown_out[12]	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
coreerrirq[12]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[12]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[12]	Signal	Slave	Disable cryptography extensions after reset.
cti[12]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[12]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[12]	Signal	Master	No power-down request.
dbgpwrupreq[12]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.



Name	Protocol	Type	Description
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[12]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[12]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[12]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmuirq[12]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[12]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[12]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[12]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[12]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[12]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[12]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[12]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[12]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[12]	Signal	Slave	Virtualised FIQ.
virq[12]	Signal	Slave	Virtualised IRQ.

Name	Protocol	Type	Description
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[12]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARM CortexX2CT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### BROADCASTOUTER

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

### BROADCASTPERSIST

#### Type

bool

#### Default value

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.



**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`dcache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`default_opmode`****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamlQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamlQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x20

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x2

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.



**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect.  
1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**`icache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or

miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

### **`icache-read_latency`**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

### **`icache-state_modelled`**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Set whether I-cache has stateful implementation.

### **`invalidate_code_cache_on_icache_cmo`**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

### **`l3cache-hit_latency`**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x0

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x2

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0



**Description**

Register core as ON state to cluster with simulation reset.

**`periph_address_end`****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**`periph_address_start`****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**`ptw_latency`****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**`tlb_latency`****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**`tlbi_stall_enabled`****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**`treat-dcache-cmos-to-pou-as-nop`**

**Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**`walk_cache_latency`**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.73 ARMCortexX3CT**

ARMCortexX3CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-477: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexX3CT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- DynamIQ r3p0.

- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.
- Utility bus.
- Support for Interrupt signals from SPE is added as pmbirq[8].

Support for the following features is planned for a future release:

- TRBE.
- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (TLM2 extensions from Arm) bus protocols.
- Core-Complex.
- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.



Note

The `cfgsdisable` signal will be removed in a future release.

---

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Limitations

Some of the SVE instructions do not raise undefined exceptions when SP is used as a source register.

## Iris and MTI instances for ARMCortexX3CT

This model has the following Iris instances:

**Table 3-478: ARMCortexX3CT Iris instances**

InstanceName	ComponentName
ARMCortexX3CT	Cluster_ARM_Cortex-X3
ARMCortexX3CT.AMU	PVBusLogger
ARMCortexX3CT.AMU.mapper	PVBusMapper
ARMCortexX3CT.DAP	PVBusLogger
ARMCortexX3CT.DAP.mapper	PVBusMapper
ARMCortexX3CT.DSU	DSU
ARMCortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX3CT.DSU.PPU_core0	PPUv1
ARMCortexX3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX3CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX3CT.DSU.shared_cache	PVCache
ARMCortexX3CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX3CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX3CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX3CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX3CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX3CT.MMAP	PVBusLogger
ARMCortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexX3CT.RAS	PVBusLogger
ARMCortexX3CT.RAS.mapper	PVBusMapper
ARMCortexX3CT.cpu0	ARM_Cortex-X3
ARMCortexX3CT.cpu0.UTLB	TLB
ARMCortexX3CT.cpu0.debug_rom	debug_rom
ARMCortexX3CT.cpu0.dtlb	TLB
ARMCortexX3CT.cpu0.l1dcache	PVCache
ARMCortexX3CT.cpu0.l1dcache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexX3CT.cpu0.l1icache	PVCache
ARMCortexX3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l2cache	PVCache
ARMCortexX3CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX3CT.ext_bus	PVBusLogger
ARMCortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexX3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX3CT.global_debug_rom	debug_rom
ARMCortexX3CT.secondary_debug_rom	debug_rom
ARMCortexX3CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-479: ARMCortexX3CT MTI instances**

InstanceName	ComponentName
ARMCortexX3CT.AMU	PVBusLogger
ARMCortexX3CT.AMU.mapper	PVBusMapper
ARMCortexX3CT.DAP	PVBusLogger
ARMCortexX3CT.DAP.mapper	PVBusMapper
ARMCortexX3CT.DSU	DSU
ARMCortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX3CT.DSU.PPU_core0	PPUv1
ARMCortexX3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX3CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX3CT.DSU.shared_cache	PVCache
ARMCortexX3CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX3CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX3CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX3CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX3CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX3CT.MMAP	PVBusLogger
ARMCortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexX3CT.RAS	PVBusLogger
ARMCortexX3CT.RAS.mapper	PVBusMapper
ARMCortexX3CT.cpu0	ARM_Cortex-X3
ARMCortexX3CT.cpu0.UTLB	TLB
ARMCortexX3CT.cpu0.l1dcache	PVCache

InstanceName	ComponentName
ARMCortexX3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l1icache	PVCache
ARMCortexX3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l2cache	PVCache
ARMCortexX3CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX3CT.ext_bus	PVBusLogger
ARMCortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexX3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexX3CT

**Table 3-480: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[12]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal

Name	Protocol	Type	Description
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[12]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[12]	Signal	Master	Timer signals to SOC
CNTHVIRQ[12]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[12]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[12]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[12]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[12]	Signal	Master	Timer signals to SOC.
commirq[12]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[12]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[12]	PChannel	Master	Core PCSM signals
core_powerdown_out[12]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[12]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[12]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[12]	Signal	Slave	Disable cryptography extensions after reset.
cti[12]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[12]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[12]	Signal	Master	No power-down request.
dbgpwrupreq[12]	Signal	Master	Debug power up request.

Name	Protocol	Type	Description
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[12]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[12]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[12]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[12]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[12]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[12]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[12]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[12]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[12]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[12]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[12]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[12]	Signal	Master	Interrupt signal from the trace buffer unit.



Name	Protocol	Type	Description
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[12]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[12]	Signal	Slave	Virtualised FIQ.
virq[12]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[12]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexX3CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAJINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-size****Type**

int

**Default value**

0x100000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0



**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**error\_record\_feature\_register****Type**

string

**Default value**

```
"[{\"ED\":0x2,\"IMPDEF_3_2\":0x0,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,
  \"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0,\"Visibility\": \"Cluster
  \",{\"ED\":0x2,\"IMPDEF_3_2\":0x0,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,
  \"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0}]]"
```

**Description**

RAS feature register values. An array of JSON objects. The JSON schema for the array is:  
 [{"ED":0x0, "IMPDEF\_3\_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility": "Core"}, other\_feature\_register\_values].



Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4. RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

**ete.CLAIMTAGS****Type**

int

**Default value**

0x20

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**ext\_abort\_normal\_noncacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of normal noncacheable-read external aborts.

**ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x2

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**`force_zero_mpam_partid_and_pmg`****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**`has_enhanced_pan`****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**`has_ete`****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**`has_large_va`****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA).

Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**instruction\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-has\_mpam****Type**

bool

**Default value**

0x1

**Description**

L3 Cache has MPAM support.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0



**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`l3cache-size`****Type**

int

**Default value**

0x40000

**Description**

L3 Cache size in bytes.

**`l3cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`l3cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`l3cache-ways`****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases

where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

### **num\_nodes**

#### **Type**

int

#### **Default value**

0x1

#### **Description**

Number of transport nodes. Zero implies direct-connect configuration.

### **pmu-num\_counters**

#### **Type**

int

#### **Default value**

0x6

#### **Description**

Number of PMU counters implemented.

### **pseudo\_fault\_generation\_feature\_register**

#### **Type**

string

#### **Default value**

```
"[{\"OF\":true, \"UC\":true, \"UEU\":false, \"UER\":false, \"UEO\":false, \"DE\":0x1, \"CE\":0x1,
\"CI\":true, \"ER\":false, \"PN\":true, \"AV\":false, \"MV\":true, \"SYN\":true, \"R\":true},{\"OF
\":false, \"UC\":true, \"UEU\":false, \"UER\":false, \"UEO\":false, \"DE\":0x1, \"CE\":0x1, \"CI
\":false, \"ER\":false, \"PN\":false, \"AV\":false, \"MV\":false, \"SYN\":false, \"R\":true}]"
```

#### **Description**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_pseudo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

### **ptw\_latency**

#### **Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**trace\_physical\_registers\_when\_host\_virtualisation\_enabled****Type**

int

**Default value**

0x1

**Description**

When host virtualisation is enabled, trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers.

**treat\_PAC\_as\_NOP**

**Type**  
bool

**Default value**  
0x0

**Description**  
Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency**

**Type**  
int

**Default value**  
0x0

**Description**  
Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.74 ARMCortexX4CT

ARMCortexX4CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-481: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMCortexX4CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- DynamIQ r3p0.
- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.

- Per-core clock.
- `BROADCASTPERSIST` pin.
- Utility bus.
- Support for Interrupt signals from SPE is added as `pmbirq[8]`.

Support for the following features is planned for a future release:

- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (TLM2 extensions from Arm) bus protocols.
- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, and `nPMBIRQ` signals.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not supported but signals are implemented.



The `cfgsdisable` signal will be removed in a future release.

---

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Limitations

Some of the SVE instructions do not raise undefined exceptions when SP is used as a source register.

## Iris and MTI instances for ARMCortexX4CT

This model has the following Iris instances:

**Table 3-482: ARMCortexX4CT Iris instances**

InstanceName	ComponentName
ARMCortexX4CT	Cluster_ARM_Cortex-X4
ARMCortexX4CT.AMU	PVBusLogger
ARMCortexX4CT.AMU.mapper	PVBusMapper
ARMCortexX4CT.DAP	PVBusLogger
ARMCortexX4CT.DAP.mapper	PVBusMapper
ARMCortexX4CT.DSU	DSU
ARMCortexX4CT.DSU.PPU_cluster	PPUv1
ARMCortexX4CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX4CT.DSU.PPU_core0	PPUv1
ARMCortexX4CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX4CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX4CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX4CT.DSU.shared_cache	PVCache
ARMCortexX4CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX4CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX4CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX4CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX4CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexX4CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX4CT.MMAP	PVBusLogger
ARMCortexX4CT.MMAP.mapper	PVBusMapper
ARMCortexX4CT.RAS	PVBusLogger
ARMCortexX4CT.RAS.mapper	PVBusMapper
ARMCortexX4CT.cpu0	ARM_Cortex-X4
ARMCortexX4CT.cpu0.UTLB	TLB
ARMCortexX4CT.cpu0.debug_rom	debug_rom
ARMCortexX4CT.cpu0.dtlb	TLB
ARMCortexX4CT.cpu0.l1dcache	PVCache
ARMCortexX4CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX4CT.cpu0.l1licache	PVCache
ARMCortexX4CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexX4CT.cpu0.l2cache	PVCache
ARMCortexX4CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexX4CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX4CT.ext_bus	PVBusLogger



InstanceName	ComponentName
ARMCortexX4CT.ext_bus.mapper	PVBusMapper
ARMCortexX4CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX4CT.global_debug_rom	debug_rom
ARMCortexX4CT.secondary_debug_rom	debug_rom
ARMCortexX4CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-483: ARMCortexX4CT MTI instances**

InstanceName	ComponentName
ARMCortexX4CT.AMU	PVBusLogger
ARMCortexX4CT.AMU.mapper	PVBusMapper
ARMCortexX4CT.DAP	PVBusLogger
ARMCortexX4CT.DAP.mapper	PVBusMapper
ARMCortexX4CT.DSU	DSU
ARMCortexX4CT.DSU.PPU_cluster	PPUv1
ARMCortexX4CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX4CT.DSU.PPU_core0	PPUv1
ARMCortexX4CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX4CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX4CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX4CT.DSU.shared_cache	PVCache
ARMCortexX4CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX4CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX4CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX4CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX4CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexX4CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX4CT.MMAP	PVBusLogger
ARMCortexX4CT.MMAP.mapper	PVBusMapper
ARMCortexX4CT.RAS	PVBusLogger
ARMCortexX4CT.RAS.mapper	PVBusMapper
ARMCortexX4CT.cpu0	ARM_Cortex-X4
ARMCortexX4CT.cpu0.UTLB	TLB
ARMCortexX4CT.cpu0.l1dcache	PVCache
ARMCortexX4CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX4CT.cpu0.l1licache	PVCache
ARMCortexX4CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexX4CT.cpu0.l2cache	PVCache
ARMCortexX4CT.cpu0.l2cache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexX4CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX4CT.ext_bus	PVBusLogger
ARMCortexX4CT.ext_bus.mapper	PVBusMapper
ARMCortexX4CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexX4CT

**Table 3-484: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[14]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[14]	Signal	Master	Timer signals to SOC
CNTHVIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[14]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[14]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[14]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[14]	Signal	Master	Timer signals to SOC.
commirq[14]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[14]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[14]	PChannel	Master	Core PCSM signals
core_powerdown_out[14]	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
coreerrirq[14]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[14]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[14]	Signal	Slave	Disable cryptography extensions after reset.
cti[14]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[14]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[14]	Signal	Master	No power-down request.
dbgpwrupreq[14]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.

Name	Protocol	Type	Description
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[14]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[14]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[14]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[14]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[14]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[14]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[14]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[14]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[14]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[14]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[14]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[14]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[14]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.

Name	Protocol	Type	Description
vfiq[14]	Signal	Slave	Virtualised FIQ.
virq[14]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[14]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexX4CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

### AEND3\_DEFAULT

#### Type

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1



**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`cpuX.l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-ways`****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**`cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`cpuX.max_code_cache_mb`****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0



**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**ecv\_support\_level****Type**

int

**Default value**

0x2

**Description**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x2

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**`force_zero_mpam_partid_and_pmg`****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**`has_enhanced_pan`****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**`has_ete`****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**`has_peripheral_port`****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.



**has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**has\_v8\_7\_spe\_inverted\_filtering****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

**has\_v8\_7\_spe\_previous\_branch\_target****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**log2\_trace\_buffer\_alignment****Type**

int

**Default value**

0x6

**Description**

Log2 of trace buffer alignment constraint for output buffer (0-&gt;1B ... 11-&gt;2Kib).

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpam\_has\_altsp****Type**

bool

**Default value**

0x0

**Description**

MPAM Whether MPAMIDR\_EL1.HAS\_ALTSP bit is set or clear.

**mpamidr\_has\_force\_ns****Type**

int

**Default value**

0x0

**Description**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpamidr\_has\_sdeflt****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpamidr\_has\_tidr****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate

the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantums in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_acp****Type**

int

**Default value**

0x0

**Description**

Number of ACP ports.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pmu-num\_counters****Type**

int

**Default value**

0x1f

**Description**

Number of PMU counters implemented.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**tcr\_txsz\_undersize\_should\_fault****Type**

bool

**Default value**

0x0

**Description**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\*

instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAAuth traps.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.75 **ARMCortexX925CT**

ARMCortexX925CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-485: IP revisions support**

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMCortexX925CT**

A DSU-120 DynamIQ cluster containing a configurable number of Cortex-X925 cores.

The number of cores in the cluster is configurable using the following parameter:

**NUM\_CORES**

Possible values are 1-14

The following DSU/CPU features will not be implemented:

- DynamIQ features that are irrelevant to the programmers’ view simulation.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).



## Iris and MTI instances for ARMCortexX925CT

This model has the following Iris instances:

**Table 3-486: ARMCortexX925CT Iris instances**

InstanceName	ComponentName
ARMCortexX925CT	Cluster_ARM_Cortex-X925
ARMCortexX925CT.AMU	PVBusLogger
ARMCortexX925CT.AMU.mapper	PVBusMapper
ARMCortexX925CT.DAP	PVBusLogger
ARMCortexX925CT.DAP.mapper	PVBusMapper
ARMCortexX925CT.DSU	DSU
ARMCortexX925CT.DSU.PPU_cluster	PPUv1
ARMCortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX925CT.DSU.PPU_core0	PPUv1
ARMCortexX925CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX925CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX925CT.DSU.shared_cache	PVCache
ARMCortexX925CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX925CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX925CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX925CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX925CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX925CT.MMAP	PVBusLogger
ARMCortexX925CT.MMAP.mapper	PVBusMapper
ARMCortexX925CT.RAS	PVBusLogger
ARMCortexX925CT.RAS.mapper	PVBusMapper
ARMCortexX925CT.cpu0	ARM_Cortex-X925
ARMCortexX925CT.cpu0.UTLB	TLB
ARMCortexX925CT.cpu0.debug_rom	debug_rom
ARMCortexX925CT.cpu0.dtlb	TLB
ARMCortexX925CT.cpu0.l1dcache	PVCache
ARMCortexX925CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX925CT.cpu0.l1icache	PVCache
ARMCortexX925CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX925CT.cpu0.l2cache	PVCache
ARMCortexX925CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMCortexX925CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX925CT.ext_bus	PVBusLogger

InstanceName	ComponentName
ARMCortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX925CT.global_debug_rom	debug_rom
ARMCortexX925CT.secondary_debug_rom	debug_rom
ARMCortexX925CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-487: ARMCortexX925CT MTI instances**

InstanceName	ComponentName
ARMCortexX925CT.AMU	PVBusLogger
ARMCortexX925CT.AMU.mapper	PVBusMapper
ARMCortexX925CT.DAP	PVBusLogger
ARMCortexX925CT.DAP.mapper	PVBusMapper
ARMCortexX925CT.DSU	DSU
ARMCortexX925CT.DSU.PPU_cluster	PPUv1
ARMCortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX925CT.DSU.PPU_core0	PPUv1
ARMCortexX925CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX925CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX925CT.DSU.shared_cache	PVCache
ARMCortexX925CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMCortexX925CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMCortexX925CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMCortexX925CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMCortexX925CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMCortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX925CT.MMAP	PVBusLogger
ARMCortexX925CT.MMAP.mapper	PVBusMapper
ARMCortexX925CT.RAS	PVBusLogger
ARMCortexX925CT.RAS.mapper	PVBusMapper
ARMCortexX925CT.cpu0	ARM_Cortex-X925
ARMCortexX925CT.cpu0.UTLB	TLB
ARMCortexX925CT.cpu0.l1dcache	PVCache
ARMCortexX925CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX925CT.cpu0.l1icache	PVCache
ARMCortexX925CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX925CT.cpu0.l2cache	PVCache
ARMCortexX925CT.cpu0.l2cache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMCortexX925CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMCortexX925CT.ext_bus	PVBusLogger
ARMCortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexX925CT

**Table 3-488: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[14]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[14]	Signal	Master	Timer signals to SOC
CNTHVIRQ[14]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[14]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[14]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[14]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[14]	Signal	Master	Timer signals to SOC.
commirq[14]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[14]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[14]	PChannel	Master	Core PCSM signals
core_powerdown_out[14]	Signal	Master	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
coreerrirq[14]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[14]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[14]	Signal	Slave	Disable cryptography extensions after reset.
cti[14]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[14]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[14]	Signal	Master	No power-down request.
dbgpwrupreq[14]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.

Name	Protocol	Type	Description
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[14]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[14]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[14]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[14]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[14]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[14]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[14]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[14]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[14]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[14]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[14]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[14]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[14]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.

Name	Protocol	Type	Description
vfiq[14]	Signal	Slave	Virtualised FIQ.
virq[14]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[14]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMCortexX925CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

### AEND3\_DEFAULT

#### Type

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.



**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`cpuX.l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-ways`****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**`cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`cpuX.max_code_cache_mb`****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.



**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**ecv\_support\_level****Type**

int

**Default value**

0x2

**Description**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x2

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0



**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**`force_zero_mpam_partid_and_pmg`****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**`has_enhanced_pan`****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**`has_ete`****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**`has_mt_pmu_disable_feature`****Type**

int

**Default value**

0x0

**Description**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2:

FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**has\_v8\_7\_spe\_inverted\_filtering****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

**has\_v8\_7\_spe\_previous\_branch\_target****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**`icache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-size****Type**

int

**Default value**

0x80000

**Description**

L3 Cache size in bytes.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**log2\_trace\_buffer\_alignment****Type**

int

**Default value**

0x6

**Description**

Log2 of trace buffer alignment constraint for output buffer (0-&gt;1B ... 11-&gt;2Kib).

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpam\_has\_altsp****Type**

bool

**Default value**

0x0

**Description**

MPAM Whether MPAMIDR\_EL1.HAS\_ALTSP bit is set or clear.

**mpamidr\_has\_force\_ns****Type**

int

**Default value**

0x0

**Description**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpamidr\_has\_sdeflt****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpamidr\_has\_tidr****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear Possible values of this parameter are:  
- 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_acp****Type**

int

**Default value**

0x0

**Description**

Number of ACP ports.

**num\_nodes****Type**

int

**Default value**

0x1

**Description**

Number of transport nodes. Zero implies direct-connect configuration.

**pmu-num\_counters****Type**

int

**Default value**

0x1f

**Description**

Number of PMU counters implemented.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**tcr\_txsz\_undersize\_should\_fault****Type**

bool

**Default value**

0x0

**Description**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0



**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**`treat_PAC_as_NOP`**

**Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**`walk_cache_latency`**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.76 ARMNeoverseE1CT**

ARMNeoverseE1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-489: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMNeoverseE1CT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.

- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGWRDUP and DBGRSTREQ are not implemented, but DBGWRUPREQ and DBGNOPWRDWN are implemented.
- Cache stashing capability.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseE1CT

This model has the following Iris instances:

**Table 3-490: ARMNeoverseE1CT Iris instances**

InstanceName	ComponentName
ARMNeoverseE1CT	Cluster_ARM_Neoverse-E1
ARMNeoverseE1CT.AMU	PVBusLogger
ARMNeoverseE1CT.AMU.mapper	PVBusMapper
ARMNeoverseE1CT.DAP	PVBusLogger
ARMNeoverseE1CT.DAP.mapper	PVBusMapper

InstanceName	ComponentName
ARMNeoverseE1CT.DSU	DSU
ARMNeoverseE1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMNeoverseE1CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseE1CT.DSU.shared_cache	PVCache
ARMNeoverseE1CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseE1CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseE1CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseE1CT.MMAP	PVBusLogger
ARMNeoverseE1CT.MMAP.mapper	PVBusMapper
ARMNeoverseE1CT.RAS	PVBusLogger
ARMNeoverseE1CT.RAS.mapper	PVBusMapper
ARMNeoverseE1CT.cpu0.dtlb	TLB
ARMNeoverseE1CT.cpu0.thread0	ARM_Neoverse-E1
ARMNeoverseE1CT.cpu0.thread0.UTLB	TLB
ARMNeoverseE1CT.cpu0.thread0.l1dcache	PVCache
ARMNeoverseE1CT.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread0.l1licache	PVCache
ARMNeoverseE1CT.cpu0.thread0.l1licache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread0.l2cache	PVCache
ARMNeoverseE1CT.cpu0.thread0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread1	ARM_Neoverse-E1
ARMNeoverseE1CT.cpu0.thread1.UTLB	TLB
ARMNeoverseE1CT.ext_bus	PVBusLogger
ARMNeoverseE1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseE1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseE1CT.global_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-491: ARMNeoverseE1CT MTI instances**

InstanceName	ComponentName
ARMNeoverseE1CT.AMU	PVBusLogger
ARMNeoverseE1CT.AMU.mapper	PVBusMapper
ARMNeoverseE1CT.DAP	PVBusLogger
ARMNeoverseE1CT.DAP.mapper	PVBusMapper
ARMNeoverseE1CT.DSU	DSU
ARMNeoverseE1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMNeoverseE1CT.DSU.mpam_busslave	PVBusSlave

InstanceName	ComponentName
ARMNeoverseE1CT.DSU.shared_cache	PVCache
ARMNeoverseE1CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseE1CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseE1CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseE1CT.MMAP	PVBusLogger
ARMNeoverseE1CT.MMAP.mapper	PVBusMapper
ARMNeoverseE1CT.RAS	PVBusLogger
ARMNeoverseE1CT.RAS.mapper	PVBusMapper
ARMNeoverseE1CT.cpu0.thread0	ARM_Neoverse-E1
ARMNeoverseE1CT.cpu0.thread0.UTLB	TLB
ARMNeoverseE1CT.cpu0.thread0.l1dcache	PVCache
ARMNeoverseE1CT.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread0.l1icache	PVCache
ARMNeoverseE1CT.cpu0.thread0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread0.l2cache	PVCache
ARMNeoverseE1CT.cpu0.thread0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread1	ARM_Neoverse-E1
ARMNeoverseE1CT.cpu0.thread1.UTLB	TLB
ARMNeoverseE1CT.ext_bus	PVBusLogger
ARMNeoverseE1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseE1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMNeoverseE1CT

**Table 3-492: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[16]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.

Name	Protocol	Type	Description
cfgte[16]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[16]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[16]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[16]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[16]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[16]	Signal	Master	Timer signals to SOC.
commirq[16]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[8]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable[16]	Signal	Slave	This signal disables write access to some system control processor registers.
cpuporeset[8]	Signal	Slave	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[16]	Signal	Slave	Disable cryptography extensions after reset.
cti[16]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[16]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[16]	Signal	Master	These signals relate to core power down.
dbgpwrupreq[16]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[16]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.

Name	Protocol	Type	Description
gicv3_redistributor_s[16]	GICv3Comms	Slave	GICv3 AXI-stream port per thread.
irq[16]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[8]	PChannel	Slave	PChannels for cores
pmuirq[16]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[16]	Signal	Slave	Per core RAM Error Interrupt.
reset[8]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[16]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[16]	Signal	Slave	Per core virtual System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[16]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[16]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[16]	Signal	Slave	Virtualised FIQ.
vinithi[16]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[16]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[16]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseE1CT

### **BROADCASTATOMIC**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### **BROADCASTCACHEMAINT**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

### **BROADCASTOUTER**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

### **BROADCASTPERSIST**

**Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1



**Description**

Number of cores per cluster.

**cluster\_patch\_level****Type**

int

**Default value**

0x0

**Description**

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

**cluster\_revision\_number****Type**

int

**Default value**

0x0

**Description**

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.enable\_single\_thread\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable single thread after reset and keep other thread in reset.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-size****Type**

int

**Default value**

0x40000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.threadY.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.threadY.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.threadY.MPIDR-override****Type**

int

**Default value**

0x0

**Description**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

**cpuX.threadY.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.threadY.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.



**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_dot\_product****Type**

int

**Default value**

0x2

**Description**

Possible values of this parameter are: - 1, feature not implemented. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**l3cache-size****Type**

int

**Default value**

0x400000



**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**  
TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled**

**Type**  
bool

**Default value**  
0x0

**Description**  
If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**walk\_cache\_latency**

**Type**  
int

**Default value**  
0x0

**Description**  
Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.77 ARMNeoverseN1CT

ARMNeoverseN1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-493: IP revisions support

Revision	Quality level
r4p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMNeoverseN1CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.

- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGWRUP and DBGRSTREQ are not implemented, but DBGWRUPREQ and DBGNOPWRDN are implemented.
- Cache stashing capability.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseN1CT

This model has the following Iris instances:

**Table 3-494: ARMNeoverseN1CT Iris instances**

InstanceName	ComponentName
ARMNeoverseN1CT	Cluster_ARM_Neoverse-N1
ARMNeoverseN1CT.AMU	PVBusLogger
ARMNeoverseN1CT.AMU.mapper	PVBusMapper
ARMNeoverseN1CT.DAP	PVBusLogger
ARMNeoverseN1CT.DAP.mapper	PVBusMapper
ARMNeoverseN1CT.DSU	DSU
ARMNeoverseN1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMNeoverseN1CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN1CT.DSU.shared_cache	PVCache
ARMNeoverseN1CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseN1CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseN1CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseN1CT.MMAP	PVBusLogger

InstanceName	ComponentName
ARMNeoverseN1CT.MMAP.mapper	PVBusMapper
ARMNeoverseN1CT.RAS	PVBusLogger
ARMNeoverseN1CT.RAS.mapper	PVBusMapper
ARMNeoverseN1CT.cpu0	ARM_Neoverse-N1
ARMNeoverseN1CT.cpu0.UTLB	TLB
ARMNeoverseN1CT.cpu0.debug_rom	debug_rom
ARMNeoverseN1CT.cpu0.dtlb	TLB
ARMNeoverseN1CT.cpu0.l1dcache	PVCache
ARMNeoverseN1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.cpu0.l1icache	PVCache
ARMNeoverseN1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.cpu0.l2cache	PVCache
ARMNeoverseN1CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseN1CT.ext_bus	PVBusLogger
ARMNeoverseN1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseN1CT.global_debug_rom	debug_rom
ARMNeoverseN1CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

**Table 3-495: ARMNeoverseN1CT MTI instances**

InstanceName	ComponentName
ARMNeoverseN1CT.AMU	PVBusLogger
ARMNeoverseN1CT.AMU.mapper	PVBusMapper
ARMNeoverseN1CT.DAP	PVBusLogger
ARMNeoverseN1CT.DAP.mapper	PVBusMapper
ARMNeoverseN1CT.DSU	DSU
ARMNeoverseN1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMNeoverseN1CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN1CT.DSU.shared_cache	PVCache
ARMNeoverseN1CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseN1CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseN1CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseN1CT.MMAP	PVBusLogger
ARMNeoverseN1CT.MMAP.mapper	PVBusMapper
ARMNeoverseN1CT.RAS	PVBusLogger
ARMNeoverseN1CT.RAS.mapper	PVBusMapper

InstanceName	ComponentName
ARMNeoverseN1CT.cpu0	ARM_Neoverse-N1
ARMNeoverseN1CT.cpu0.UTLB	TLB
ARMNeoverseN1CT.cpu0.l1dcache	PVCache
ARMNeoverseN1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.cpu0.l1icache	PVCache
ARMNeoverseN1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.cpu0.l2cache	PVCache
ARMNeoverseN1CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseN1CT.ext_bus	PVBusLogger
ARMNeoverseN1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMNeoverseN1CT

**Table 3-496: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[4]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.

Name	Protocol	Type	Description
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[4]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[4]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.

Name	Protocol	Type	Description
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseN1CT

### BROADCASTATOMIC

#### Type

bool



**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CFGTE`****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**`cpuX.CRYPTODISABLE`****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**`cpuX.RVBARADDR`****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**`cpuX.semihosting-T32_HLT`****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**`cpuX.semihosting-Thumb_SVC`****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**`cpuX.semihosting-cmd_line`****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**`cpuX.semihosting-cwd`****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**`cpuX.semihosting-enable`****Type**

bool

**Default value**

0x1



**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_coherent\_icache****Type**

bool

**Default value**

0x0

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.



**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**l3cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**l3cache-size****Type**

int

**Default value**

0x100000

**Description**

L3 Cache size in bytes.

**l3cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**l3cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

**`periph_address_start`****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

**`ptw_latency`****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**`tlb_latency`****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**`tlbi_stall_enabled`****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat-dcache-cmos-to-pou-as-nop**

**Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

**3.5.78 ARMNeoverseN2CT**

ARMNeoverseN2CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-497: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMNeoverseN2CT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

Support for the following features is planned for a future release:

- DynamIQ Shared Unit-110 (DSU-110) system registers.
- BROADCASTCACHEMAINTPOU pin
- COREINSTRRET, COREINSTRRUN, and nPMBIRQ signals
- DSU-110 cluster. The implementation relies on DynamIQ only.
- TRBE.
- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (the Arm TLM2 extensions) bus protocols.

The following features will not be implemented:

- 256-bit wide output transactions.
- Error correction/detection features.
- Self-test features (MBIST).
- Snoop filtering.
- Armv9 trace extensions.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseN2CT

This model has the following Iris instances:

**Table 3-498: ARMNeoverseN2CT Iris instances**

InstanceName	ComponentName
ARMNeoverseN2CT	Cluster_ARM_Neoverse-N2
ARMNeoverseN2CT.AMU	PVBusLogger
ARMNeoverseN2CT.AMU.mapper	PVBusMapper
ARMNeoverseN2CT.DAP	PVBusLogger
ARMNeoverseN2CT.DAP.mapper	PVBusMapper
ARMNeoverseN2CT.DSU	DSU
ARMNeoverseN2CT.DSU.PPU_cluster	PPUv1
ARMNeoverseN2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseN2CT.DSU.PPU_core0	PPUv1
ARMNeoverseN2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseN2CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN2CT.DSU.shared_cache	PVCache
ARMNeoverseN2CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseN2CT.DSU.shared_cache.upstream[2]	PVBusSlave

InstanceName	ComponentName
ARMNeoverseN2CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseN2CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseN2CT.MMAP	PVBusLogger
ARMNeoverseN2CT.MMAP.mapper	PVBusMapper
ARMNeoverseN2CT.RAS	PVBusLogger
ARMNeoverseN2CT.RAS.mapper	PVBusMapper
ARMNeoverseN2CT.cpu0	ARM_Neoverse-N2
ARMNeoverseN2CT.cpu0.UTLB	TLB
ARMNeoverseN2CT.cpu0.debug_rom	debug_rom
ARMNeoverseN2CT.cpu0.dtlb	TLB
ARMNeoverseN2CT.cpu0.l1dcache	PVCache
ARMNeoverseN2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.cpu0.l1icache	PVCache
ARMNeoverseN2CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.cpu0.l2cache	PVCache
ARMNeoverseN2CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseN2CT.default_CTM	EmbeddedCT
ARMNeoverseN2CT.ext_bus	PVBusLogger
ARMNeoverseN2CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseN2CT.global_debug_rom	debug_rom
ARMNeoverseN2CT.secondary_debug_rom	debug_rom
ARMNeoverseN2CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-499: ARMNeoverseN2CT MTI instances**

InstanceName	ComponentName
ARMNeoverseN2CT.AMU	PVBusLogger
ARMNeoverseN2CT.AMU.mapper	PVBusMapper
ARMNeoverseN2CT.DAP	PVBusLogger
ARMNeoverseN2CT.DAP.mapper	PVBusMapper
ARMNeoverseN2CT.DSU	DSU
ARMNeoverseN2CT.DSU.PPU_cluster	PPUv1
ARMNeoverseN2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseN2CT.DSU.PPU_core0	PPUv1
ARMNeoverseN2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseN2CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN2CT.DSU.shared_cache	PVCache

InstanceName	ComponentName
ARMNeoverseN2CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseN2CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseN2CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseN2CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseN2CT.MMAP	PVBusLogger
ARMNeoverseN2CT.MMAP.mapper	PVBusMapper
ARMNeoverseN2CT.RAS	PVBusLogger
ARMNeoverseN2CT.RAS.mapper	PVBusMapper
ARMNeoverseN2CT.cpu0	ARM_Neoverse-N2
ARMNeoverseN2CT.cpu0.UTLB	TLB
ARMNeoverseN2CT.cpu0.l1dcache	PVCache
ARMNeoverseN2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.cpu0.l1icache	PVCache
ARMNeoverseN2CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.cpu0.l2cache	PVCache
ARMNeoverseN2CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseN2CT.ext_bus	PVBusLogger
ARMNeoverseN2CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMNeoverseN2CT

**Table 3-500: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).



Name	Protocol	Type	Description
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[1]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[1]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[1]	Signal	Master	Timer signals to SOC
CNTHVIRQ[1]	Signal	Master	Timer signals to SOC.
CNTHVSIIRQ[1]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[1]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[1]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[1]	Signal	Master	Timer signals to SOC.
commirq[1]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[1]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel[1]	PChannel	Master	Core PCSM signals

Name	Protocol	Type	Description
core_powerdown_out[1]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[1]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[1]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[1]	Signal	Slave	Disable cryptography extensions after reset.
cti[1]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[1]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[1]	Signal	Master	No power-down request.
dbgpwrupreq[1]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[1]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[1]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[1]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[1]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[1]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[1]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[1]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.

Name	Protocol	Type	Description
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[1]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[1]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[1]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[1]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[1]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[1]	Signal	Slave	Virtualised FIQ.
virq[1]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[1]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseN2CT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`**

**Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`**

**Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CRYPTODISABLE`**

**Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**`cpuX.RVBARADDR`**

**Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**`cpuX.enable_trace_special_hlt_imm16`**

**Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter `trace_special_hlt_imm16`.

**`cpuX.l2cache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x100000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int



**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**`cpuX.semihosting-A64_HLT`****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**`cpuX.semihosting-ARM_SVC`****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**`cpuX.semihosting-T32_HLT`****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**`cpuX.semihosting-Thumb_SVC`****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**`cpuX.semihosting-cmd_line`****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**`cpuX.semihosting-stack_limit`****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**`cpuX.trace_special_hlt_imm16`****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**`cpuX.vfp-enable_at_reset`****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**`dcache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x20

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.



**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_read\_ras\_type****Type**

int

**Default value**

0x0

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**`force_mte_tag_access_razwi_and_ignore_tag_checks`****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**`force_zero_PSTATE_PAN`****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**`force_zero_mpam_partid_and_pmg`****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**`has_acp`****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_coherent\_icache****Type**

bool

**Default value**

0x0

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_external\_rndr****Type**

int

**Default value**

0x1

**Description**

Implement external random number generator module. When enabling this with has\_rndr enabled, the external random number generator will be used instead of internal random number generator.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**has\_rndr****Type**

int

**Default value**

0x1

**Description**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit



or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

### **`l3cache-read_latency`**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

### **`l3cache-snoop_data_transfer_latency`**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

### **`l3cache-snoop_issue_latency`**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

### **`l3cache-write_access_latency`**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

### **l3cache-write\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

### **memory\_tagging\_support\_level**

#### **Type**

int

#### **Default value**

0x2

#### **Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

### **mpam\_max\_partid**

#### **Type**

int

#### **Default value**

0x1ff

#### **Description**

MPAM Maximum PARTID Supported.

### **mpmm\_accumulator\_multiplier**

#### **Type**

int

#### **Default value**

0x1

#### **Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**rndr\_rndrrs\_seed****Type**

int

**Default value**

0x0

**Description**

Initial seed for random engine used in RNDR register.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.79 ARMNeoverseN3CT

ARMNeoverseN3CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-501: IP revisions support**

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About ARMNeoverseN3CT

A DSU-120 DynamIQ cluster containing a single Neoverse-N3 core configured for Direct connect.

The following DSU/CPU features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

#### Iris and MTI instances for ARMNeoverseN3CT

This model has the following Iris instances:

**Table 3-502: ARMNeoverseN3CT Iris instances**

InstanceName	ComponentName
ARMNeoverseN3CT	Cluster_ARM_Neoverse-N3
ARMNeoverseN3CT.AMU	PVBusLogger
ARMNeoverseN3CT.AMU.mapper	PVBusMapper
ARMNeoverseN3CT.DAP	PVBusLogger
ARMNeoverseN3CT.DAP.mapper	PVBusMapper
ARMNeoverseN3CT.DSU	DSU
ARMNeoverseN3CT.DSU.PPU_cluster	PPUv1
ARMNeoverseN3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseN3CT.DSU.PPU_core0	PPUv1

InstanceName	ComponentName
ARMNeoverseN3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseN3CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache	PVCache
ARMNeoverseN3CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMNeoverseN3CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseN3CT.MMAP	PVBusLogger
ARMNeoverseN3CT.MMAP.mapper	PVBusMapper
ARMNeoverseN3CT.RAS	PVBusLogger
ARMNeoverseN3CT.RAS.mapper	PVBusMapper
ARMNeoverseN3CT.cpu0	ARM_Neoverse-N3
ARMNeoverseN3CT.cpu0.UTLB	TLB
ARMNeoverseN3CT.cpu0.debug_rom	debug_rom
ARMNeoverseN3CT.cpu0.dtlb	TLB
ARMNeoverseN3CT.cpu0.l1dcache	PVCache
ARMNeoverseN3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.cpu0.l1icache	PVCache
ARMNeoverseN3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.cpu0.l2cache	PVCache
ARMNeoverseN3CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseN3CT.default_CTM	EmbeddedCT
ARMNeoverseN3CT.ext_bus	PVBusLogger
ARMNeoverseN3CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseN3CT.global_debug_rom	debug_rom
ARMNeoverseN3CT.secondary_debug_rom	debug_rom
ARMNeoverseN3CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-503: ARMNeoverseN3CT MTI instances**

InstanceName	ComponentName
ARMNeoverseN3CT.AMU	PVBusLogger
ARMNeoverseN3CT.AMU.mapper	PVBusMapper
ARMNeoverseN3CT.DAP	PVBusLogger
ARMNeoverseN3CT.DAP.mapper	PVBusMapper

InstanceName	ComponentName
ARMNeoverseN3CT.DSU	DSU
ARMNeoverseN3CT.DSU.PPU_cluster	PPUv1
ARMNeoverseN3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseN3CT.DSU.PPU_core0	PPUv1
ARMNeoverseN3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseN3CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache	PVCache
ARMNeoverseN3CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMNeoverseN3CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseN3CT.MMAP	PVBusLogger
ARMNeoverseN3CT.MMAP.mapper	PVBusMapper
ARMNeoverseN3CT.RAS	PVBusLogger
ARMNeoverseN3CT.RAS.mapper	PVBusMapper
ARMNeoverseN3CT.cpu0	ARM_Neoverse-N3
ARMNeoverseN3CT.cpu0.UTLB	TLB
ARMNeoverseN3CT.cpu0.l1dcache	PVCache
ARMNeoverseN3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.cpu0.l1icache	PVCache
ARMNeoverseN3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.cpu0.l2cache	PVCache
ARMNeoverseN3CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseN3CT.ext_bus	PVBusLogger
ARMNeoverseN3CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMNeoverseN3CT

**Table 3-504: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).

Name	Protocol	Type	Description
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[1]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[1]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[1]	Signal	Master	Timer signals to SOC
CNTHVIRQ[1]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[1]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[1]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[1]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.



Name	Protocol	Type	Description
CNTVIRQ[1]	Signal	Master	Timer signals to SOC.
commirq[1]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[1]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[1]	PChannel	Master	Core PCSM signals
core_powerdown_out[1]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[1]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[1]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[1]	Signal	Slave	Disable cryptography extensions after reset.
cti[1]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[1]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[1]	Signal	Master	No power-down request.
dbgpwrupreq[1]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[1]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[1]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[1]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[1]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[1]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.

Name	Protocol	Type	Description
ppu_core_irq[1]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[1]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[1]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[1]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[1]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[1]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[1]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[1]	Signal	Slave	Virtualised FIQ.
virq[1]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[1]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseN3CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

**AEND1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

**AEND2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x4

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0



**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-size`****Type**

int

**Default value**

0x100000

**Description**

L2 Cache size in bytes.

**`cpuX.l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**  
A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC**

**Type**  
int

**Default value**  
0x123456

**Description**  
A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT**

**Type**  
int

**Default value**  
0x3c

**Description**  
T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC**

**Type**  
int

**Default value**  
0xab

**Description**  
T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line**

**Type**  
string

**Default value**  
""

**Description**  
Command line available to semihosting calls.

**cpuX.semihosting-cwd**

**Type**  
string

**Default value**  
""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**`cpuX.trace_special_hlt_imm16`****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

**`cpuX.vfp-enable_at_reset`****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**`cpuX.vfp-present`****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**`dcache-hit_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int



**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`dcache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`default_opmode`****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**`diagnostics`****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**`enable_simulation_performance_optimizations`****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.ETE\_REVISION****Type**

int

**Default value**

0x1

**Description**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x1

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.

**ete.TSMARK****Type**

bool

**Default value**

0x1

**Description**

Whether timestamp markers are supported.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**`force_zero_mpam_partid_and_pmg`****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**`has_coherent_icache`****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**`has_enhanced_pan`****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**`has_ete`****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**has\_rndr\_trap****Type**

int

**Default value**

0x2

**Description**

Implement trapping for RNDR and RNDRSS random number registers from ARMv8.8. (FEAT\_RNG\_TRAP) Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.8 is enabled. - 2, feature is implemented.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x0

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

**`icache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0



**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.80 ARMNeoverseV1CT

ARMNeoverseV1CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-505: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ARMNeoverseV1CT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

Support for the following features is planned for a future release:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, and nPMBIRQ signals.
- A common cache that is shared by all threads of the core. Currently, each thread has its own L1 cache and L2 cache.

- Per-thread parameters, although signals are implemented.

The following features will not be implemented:

- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- Latency configuration.
- Cache stashing capability.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseV1CT

This model has the following Iris instances:

**Table 3-506: ARMNeoverseV1CT Iris instances**

InstanceName	ComponentName
ARMNeoverseV1CT	Cluster_ARM_Neoverse-V1
ARMNeoverseV1CT.AMU	PVBusLogger
ARMNeoverseV1CT.AMU.mapper	PVBusMapper
ARMNeoverseV1CT.DAP	PVBusLogger
ARMNeoverseV1CT.DAP.mapper	PVBusMapper
ARMNeoverseV1CT.DSU	DSU
ARMNeoverseV1CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV1CT.DSU.shared_cache	PVCache
ARMNeoverseV1CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseV1CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseV1CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseV1CT.MMAP	PVBusLogger
ARMNeoverseV1CT.MMAP.mapper	PVBusMapper
ARMNeoverseV1CT.RAS	PVBusLogger
ARMNeoverseV1CT.RAS.mapper	PVBusMapper
ARMNeoverseV1CT.cpu0	ARM_Neoverse-V1
ARMNeoverseV1CT.cpu0.UTLB	TLB

InstanceName	ComponentName
ARMNeoverseV1CT.cpu0.debug_rom	debug_rom
ARMNeoverseV1CT.cpu0.dtlb	TLB
ARMNeoverseV1CT.cpu0.l1dcache	PVCache
ARMNeoverseV1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.cpu0.l1icache	PVCache
ARMNeoverseV1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.cpu0.l2cache	PVCache
ARMNeoverseV1CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseV1CT.ext_bus	PVBusLogger
ARMNeoverseV1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseV1CT.global_debug_rom	debug_rom
ARMNeoverseV1CT.secondary_debug_rom	debug_rom
ARMNeoverseV1CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-507: ARMNeoverseV1CT MTI instances**

InstanceName	ComponentName
ARMNeoverseV1CT.AMU	PVBusLogger
ARMNeoverseV1CT.AMU.mapper	PVBusMapper
ARMNeoverseV1CT.DAP	PVBusLogger
ARMNeoverseV1CT.DAP.mapper	PVBusMapper
ARMNeoverseV1CT.DSU	DSU
ARMNeoverseV1CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV1CT.DSU.shared_cache	PVCache
ARMNeoverseV1CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseV1CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseV1CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseV1CT.MMAP	PVBusLogger
ARMNeoverseV1CT.MMAP.mapper	PVBusMapper
ARMNeoverseV1CT.RAS	PVBusLogger
ARMNeoverseV1CT.RAS.mapper	PVBusMapper
ARMNeoverseV1CT.cpu0	ARM_Neoverse-V1
ARMNeoverseV1CT.cpu0.UTLB	TLB
ARMNeoverseV1CT.cpu0.l1dcache	PVCache
ARMNeoverseV1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.cpu0.l1icache	PVCache

InstanceName	ComponentName
ARMNeoverseV1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.cpu0.l2cache	PVCache
ARMNeoverseV1CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseV1CT.ext_bus	PVBusLogger
ARMNeoverseV1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMNeoverseV1CT

**Table 3-508: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AENDMP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[4]	Signal	Slave	This signal if for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
cfgte[4]	Signal	Slave	This signal provides default exception handling state.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[4]	Signal	Master	Timer signals to SOC.
CNTHVIRQ[4]	Signal	Master	Timer signals to SOC.

Name	Protocol	Type	Description
CNTPNSIRQ[4]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[4]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[4]	Signal	Master	Timer signals to SOC.
commirq[4]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[4]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset[4]	Signal	Slave	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable[4]	Signal	Slave	Disable cryptography extensions after reset.
cti[4]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[4]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[4]	Signal	Master	No power-down request.
dbgpwrupreq[4]	Signal	Master	Debug power up request.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[4]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s[4]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[4]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
niden	Signal	Slave	External debug interface.
pchannel_cluster	PChannel	Slave	PChannel for cluster.
pchannel_core[4]	PChannel	Slave	PChannels for cores
pmbirq[4]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[4]	Signal	Master	Interrupt signal from performance monitoring unit.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[4]	Signal	Slave	Per core RAM Error Interrupt.



Name	Protocol	Type	Description
reset[4]	Signal	Slave	Raising this signal will put the core into reset mode.
rvbaraddr[4]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[4]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
spniden	Signal	Slave	External debug interface.
sporeset	Signal	Slave	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks[4]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq[4]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[4]	Signal	Slave	Virtualised FIQ.
vinithi[4]	Signal	Slave	This signal controls of the location of the exception vectors at reset.
virq[4]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[4]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseV1CT

### BROADCASTATOMIC

#### Type

bool

#### Default value

0x1

#### Description

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

### BROADCASTCACHEMAINT

#### Type

bool

#### Default value

0x1

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.VINITHI****Type**

bool

**Default value**

0x0

**Description**

Reset value of SCTLR.V.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-size`****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**`cpuX.l2cache-snoop_data_transfer_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`cpuX.l2cache-snoop_issue_latency`****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.



**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`dcache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit

or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**dcache-size****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**enhanced\_pac2\_level****Type**

int

**Default value**

0x1

**Description**

Implements Enhanced PAC2 from ARMv8.6 (FEAT\_PAuth2), and PAC enhancements from ARMv9.5 (FEAT\_PAuth\_LR). options 0-3 of this feature are mandatory for ARMv8.6 but can be cherry-picked to a ARMv8.3(or greater) implementation. 0: No EnhancedPAC2, 1: EnhancedPAC2 Only (FEAT\_PAuth2), 2: EnhancedPAC2 with FPAC (FEAT\_FPAC), 3: EnhancedPAC2 with FPACCombined (FEAT\_FPACCOMBINE), 4: EnhancedPAC2 with LR signing (FEAT\_PAuth\_LR).

**ext\_abort\_device\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE read external aborts.

**ext\_abort\_device\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRE write external aborts.

**ext\_abort\_so\_read\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE read external aborts.

**ext\_abort\_so\_write\_is\_sync****Type**

bool

**Default value**

0x0

**Description**

Synchronous reporting of device-nGnRnE write external aborts.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**has\_acp****Type**

bool

**Default value**

0x0

**Description**

If true, Accelerator Coherency Port is configured.

**has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_external\_rndr****Type**

int

**Default value**

0x1

**Description**

Implement external random number generator module. When enabling this with has\_rndr enabled, the external random number generator will be used instead of internal random number generator.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, additional AXI peripheral port is configured.

**has\_rndr****Type**

int

**Default value**

0x1

**Description**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG).



**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**l3cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**l3cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**l3cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**l3cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit

or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

### **l3cache-read\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

### **l3cache-size**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache size in bytes.

### **l3cache-snoop\_data\_transfer\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

### **l3cache-snoop\_issue\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

**l3cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**l3cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**pchannel\_treat\_simreset\_as\_poreset****Type**

bool

**Default value**

0x0

**Description**

Register core as ON state to cluster with simulation reset.

**periph\_address\_end****Type**

int

**Default value**

0x0

**Description**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

**periph\_address\_start****Type**

int

**Default value**

0x0

**Description**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat-dcache-cmos-to-pou-as-nop****Type**

int

**Default value**

0x0

**Description**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

**walk\_cache\_latency**

**Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.81 ARMNeoverseV2CT

ARMNeoverseV2CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-509: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About ARMNeoverseV2CT

The model supports the following features:

- DynamIQ r3p0.
- DynamIQ Shared Unit-110 (DSU-110) system registers.
- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- PChannel for the cluster and for each core.
- BROADCASTPERSIST pin.
- Optional peripheral port.
- L3Cache partition.
- Per-core clock.
- Utility bus.

Support for the following features is planned for a future release:

- Transporting architectural metadata tags over TLM2 (OSCI) and AMBA-PV (the Arm TLM2 extensions) bus protocols.

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, and nPMBIRQ signals.
- Core-Complex.
- Each thread currently has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- Error correction or detection features.
- Self-test features (MBIST).
- Snoop filtering.
- Latency configuration
- Cache stashing capability
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseV2CT

This model has the following Iris instances:

**Table 3-510: ARMNeoverseV2CT Iris instances**

InstanceName	ComponentName
ARMNeoverseV2CT	Cluster_ARM_Neoverse-V2
ARMNeoverseV2CT.AMU	PVBusLogger
ARMNeoverseV2CT.AMU.mapper	PVBusMapper
ARMNeoverseV2CT.DAP	PVBusLogger
ARMNeoverseV2CT.DAP.mapper	PVBusMapper
ARMNeoverseV2CT.DSU	DSU
ARMNeoverseV2CT.DSU.PPU_cluster	PPUv1
ARMNeoverseV2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV2CT.DSU.PPU_core0	PPUv1
ARMNeoverseV2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV2CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV2CT.DSU.shared_cache	PVCache



InstanceName	ComponentName
ARMNeoverseV2CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseV2CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseV2CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseV2CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV2CT.MMAP	PVBusLogger
ARMNeoverseV2CT.MMAP.mapper	PVBusMapper
ARMNeoverseV2CT.RAS	PVBusLogger
ARMNeoverseV2CT.RAS.mapper	PVBusMapper
ARMNeoverseV2CT.cpu0	ARM_Neoverse-V2
ARMNeoverseV2CT.cpu0.UTLB	TLB
ARMNeoverseV2CT.cpu0.debug_rom	debug_rom
ARMNeoverseV2CT.cpu0.dtlb	TLB
ARMNeoverseV2CT.cpu0.l1dcache	PVCache
ARMNeoverseV2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.cpu0.l1icache	PVCache
ARMNeoverseV2CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.cpu0.l2cache	PVCache
ARMNeoverseV2CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseV2CT.default_CTM	EmbeddedCT
ARMNeoverseV2CT.ext_bus	PVBusLogger
ARMNeoverseV2CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseV2CT.global_debug_rom	debug_rom
ARMNeoverseV2CT.secondary_debug_rom	debug_rom
ARMNeoverseV2CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-511: ARMNeoverseV2CT MTI instances**

InstanceName	ComponentName
ARMNeoverseV2CT.AMU	PVBusLogger
ARMNeoverseV2CT.AMU.mapper	PVBusMapper
ARMNeoverseV2CT.DAP	PVBusLogger
ARMNeoverseV2CT.DAP.mapper	PVBusMapper
ARMNeoverseV2CT.DSU	DSU
ARMNeoverseV2CT.DSU.PPU_cluster	PPUv1
ARMNeoverseV2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV2CT.DSU.PPU_core0	PPUv1

InstanceName	ComponentName
ARMNeoverseV2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV2CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV2CT.DSU.shared_cache	PVCache
ARMNeoverseV2CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseV2CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseV2CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseV2CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV2CT.MMAP	PVBusLogger
ARMNeoverseV2CT.MMAP.mapper	PVBusMapper
ARMNeoverseV2CT.RAS	PVBusLogger
ARMNeoverseV2CT.RAS.mapper	PVBusMapper
ARMNeoverseV2CT.cpu0	ARM_Neoverse-V2
ARMNeoverseV2CT.cpu0.UTLB	TLB
ARMNeoverseV2CT.cpu0.l1dcache	PVCache
ARMNeoverseV2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.cpu0.l1icache	PVCache
ARMNeoverseV2CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.cpu0.l2cache	PVCache
ARMNeoverseV2CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseV2CT.ext_bus	PVBusLogger
ARMNeoverseV2CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMNeoverseV2CT

**Table 3-512: Ports**

Name	Protocol	Type	Description
acp_s	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).

Name	Protocol	Type	Description
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[1]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsmp_channel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[1]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[1]	Signal	Master	Timer signals to SOC
CNTHVIRQ[1]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[1]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[1]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[1]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[1]	Signal	Master	Timer signals to SOC.
commirq[1]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[1]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.

Name	Protocol	Type	Description
core_pcs_m_pchannel[1]	PChannel	Master	Core PCSM signals
core_powerdown_out[1]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[1]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[1]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[1]	Signal	Slave	Disable cryptography extensions after reset.
cti[1]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[1]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[1]	Signal	Master	No power-down request.
dbgpwrupreq[1]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[1]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[1]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[1]	Signal	Slave	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[1]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[1]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[1]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[1]	Signal	Slave	PPU Core wake request signals.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.

Name	Protocol	Type	Description
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[1]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rvbaraddr[1]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[1]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[1]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[1]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[1]	Signal	Slave	Virtualised FIQ.
virq[1]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[1]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseV2CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

**Description**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

**AEND2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .



**CPUCFR****Type**

int

**Default value**

0x0

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CRYPTODISABLE`****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**`cpuX.RVBARADDR`****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.force-fpsid****Type**

bool

**Default value**

0x1

**Description**

Override the FPSID value.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x100000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**`cpuX.semihosting-T32_HLT`****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**`cpuX.semihosting-Thumb_SVC`****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**`cpuX.semihosting-cmd_line`****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**`cpuX.semihosting-cwd`****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**`cpuX.semihosting-enable`****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000



**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

**dcache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**dcache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**dcache-size****Type**

int

**Default value**

0x8000

**Description**

L1 D-Cache size in bytes.

**dcache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**dcache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x20

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.PIDR\_REVAND****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVAND value.

**ete.PIDR\_REVISION****Type**

int

**Default value**

0x0

**Description**

TRCPIDR REVISION value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.REVISION****Type**

int

**Default value**

0x0

**Description**

TRCIDR1 revision value.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCSRSTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCSRSTA value for a forcibly traced exception.

**ext\_abort\_normal\_noncacheable\_read\_is\_sync****Type**

bool

**Default value**

0x1

**Description**

Synchronous reporting of normal noncacheable-read external aborts.

**ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x2

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance



of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

### **has\_coherent\_icache**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

### **has\_enhanced\_pan**

#### **Type**

int

#### **Default value**

0x2

#### **Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

### **has\_ete**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

### **has\_external\_rndr**

#### **Type**

int

#### **Default value**

0x1

#### **Description**

Implement external random number generator module. When enabling this with has\_rndr enabled, the external random number generator will be used instead of internal random number generator.

**has\_large\_va****Type**

int

**Default value**

0x0

**Description**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**has\_rndr****Type**

int

**Default value**

0x1

**Description**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

**`icache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

**`icache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

**`icache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x8000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpam\_max\_partid****Type**

int

**Default value**

0x1ff

**Description**

MPAM Maximum PARTID Supported.

**mpam\_max\_vpmr****Type**

int

**Default value**

0x7

**Description**

MPAM Maximum VPMR Supported.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**pmu-num\_counters****Type**

int

**Default value**

0x6

**Description**

Number of PMU counters implemented.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**rndr\_rndrrs\_seed****Type**

int

**Default value**

0x0

**Description**

Initial seed for random engine used in RNDR register.

**stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**trace\_physical\_registers\_when\_host\_virtualisation\_enabled****Type**

int

**Default value**

0x1

**Description**

When host virtualisation is enabled, trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers.

**treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.82 ARMNeoverseV3AECT

ARMNeoverseV3AECT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-513: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About ARMNeoverseV3AECT

A DSU-120 DynamIQ cluster containing a single Neoverse-V3AE core configured for Direct connect.

The core supports the following optional features:

- Realm Management Extension
- Coherent instruction cache

The following DSU/CPU features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

#### AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.
- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.



As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

## Iris and MTI instances for ARMNeoverseV3AECT

This model has the following Iris instances:

**Table 3-514: ARMNeoverseV3AECT Iris instances**

InstanceName	ComponentName
ARMNeoverseV3AECT	Cluster_ARM_Neoverse-V3AE
ARMNeoverseV3AECT.AMU	PVBusLogger
ARMNeoverseV3AECT.AMU.mapper	PVBusMapper
ARMNeoverseV3AECT.DAP	PVBusLogger
ARMNeoverseV3AECT.DAP.mapper	PVBusMapper
ARMNeoverseV3AECT.DSU	DSU
ARMNeoverseV3AECT.DSU.PPU_cluster	PPUv1
ARMNeoverseV3AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.PPU_core0	PPUv1
ARMNeoverseV3AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache	PVCache
ARMNeoverseV3AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMNeoverseV3AECT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV3AECT.MMAP	PVBusLogger
ARMNeoverseV3AECT.MMAP.mapper	PVBusMapper
ARMNeoverseV3AECT.RAS	PVBusLogger
ARMNeoverseV3AECT.RAS.mapper	PVBusMapper
ARMNeoverseV3AECT.cpu0	ARM_Neoverse-V3AE
ARMNeoverseV3AECT.cpu0.UTLB	TLB
ARMNeoverseV3AECT.cpu0.debug_rom	debug_rom
ARMNeoverseV3AECT.cpu0.dtlb	TLB
ARMNeoverseV3AECT.cpu0.l1dcache	PVCache
ARMNeoverseV3AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV3AECT.cpu0.l1licache	PVCache
ARMNeoverseV3AECT.cpu0.l1licache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMNeoverseV3AECT.cpu0.l2cache	PVCache
ARMNeoverseV3AECT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseV3AECT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseV3AECT.default_CTM	EmbeddedCT
ARMNeoverseV3AECT.ext_bus	PVBusLogger
ARMNeoverseV3AECT.ext_bus.mapper	PVBusMapper
ARMNeoverseV3AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseV3AECT.global_debug_rom	debug_rom
ARMNeoverseV3AECT.secondary_debug_rom	debug_rom
ARMNeoverseV3AECT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-515: ARMNeoverseV3AECT MTI instances**

InstanceName	ComponentName
ARMNeoverseV3AECT.AMU	PVBusLogger
ARMNeoverseV3AECT.AMU.mapper	PVBusMapper
ARMNeoverseV3AECT.DAP	PVBusLogger
ARMNeoverseV3AECT.DAP.mapper	PVBusMapper
ARMNeoverseV3AECT.DSU	DSU
ARMNeoverseV3AECT.DSU.PPU_cluster	PPUv1
ARMNeoverseV3AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.PPU_core0	PPUv1
ARMNeoverseV3AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache	PVCache
ARMNeoverseV3AECT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMNeoverseV3AECT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV3AECT.MMAP	PVBusLogger
ARMNeoverseV3AECT.MMAP.mapper	PVBusMapper
ARMNeoverseV3AECT.RAS	PVBusLogger
ARMNeoverseV3AECT.RAS.mapper	PVBusMapper
ARMNeoverseV3AECT.cpu0	ARM_Neoverse-V3AE
ARMNeoverseV3AECT.cpu0.UTLB	TLB
ARMNeoverseV3AECT.cpu0.l1dcache	PVCache
ARMNeoverseV3AECT.cpu0.l1dcache.upstream[0]	PVBusSlave

InstanceName	ComponentName
ARMNeoverseV3AECT.cpu0.l1icache	PVCache
ARMNeoverseV3AECT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV3AECT.cpu0.l2cache	PVCache
ARMNeoverseV3AECT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseV3AECT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseV3AECT.ext_bus	PVBusLogger
ARMNeoverseV3AECT.ext_bus.mapper	PVBusMapper
ARMNeoverseV3AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMNeoverseV3AECT

**Table 3-516: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[1]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	PChannel	Master	Cluster PCSN signal

Name	Protocol	Type	Description
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[1]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[1]	Signal	Master	Timer signals to SOC
CNTHVIRQ[1]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[1]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[1]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[1]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[1]	Signal	Master	Timer signals to SOC.
commirq[1]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[1]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel[1]	PChannel	Master	Core PCSM signals
core_powerdown_out[1]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[1]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[1]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[1]	Signal	Slave	Disable cryptography extensions after reset.
cti[1]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[1]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[1]	Signal	Master	No power-down request.
dbgpwrupreq[1]	Signal	Master	Debug power up request.

Name	Protocol	Type	Description
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[1]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[1]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[1]	Signal	Slave	This signal drives the CPUs interrupt handling.
l0gptsz	Value	Slave	RME LOGPTSZ port
legacy_tz_en	Signal	Slave	RME LEGACY_TZ_EN port
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[1]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[1]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[1]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[1]	Signal	Slave	PPU core wake request signal.
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[1]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rlpiden	Signal	Slave	External debug interface.
rtpiden	Signal	Slave	External debug interface.
rvbaraddr[1]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[1]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.

Name	Protocol	Type	Description
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[1]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[1]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[1]	Signal	Slave	Virtualised FIQ.
virq[1]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[1]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseV3AECT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

### AEND1\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

### AEND2\_DEFAULT

#### Type

int

#### Default value

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool



**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x20

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**brbe\_log2\_num\_records****Type**

int

**Default value**

0x5

**Description**

Log2 of number of BRB records supported. 3 -> 8 records, ... 6 -> 64 records.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**`cpi_div`****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**`cpi_mul`****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**`cpuX.CFGEND`****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**`cpuX.CFGTE`****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**`cpuX.CRYPTODISABLE`****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.crypto\_aes****Type**

int

**Default value**

0x2

**Description**

AES instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 2, AES and PMULL instructions implemented (FEAT\_AES, FEAT\_PMULL).

**cpuX.crypto\_sha3****Type**

int

**Default value**

0x2

**Description**

Implement ARMv8.4 SHA-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA3). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.crypto\_sha512****Type**

int

**Default value**

0x2

**Description**

Implement ARMv8.4 SHA-512 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA512). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.crypto\_sm3****Type**

int

**Default value**

0x2

**Description**

Implement ARMv8.4 SM-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM3). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.crypto\_sm4****Type**

int

**Default value**

0x2

**Description**

Implement ARMv8.4 SM-4 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM4). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool

**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-size****Type**

int

**Default value**

0x80000

**Description**

L2 Cache size in bytes.

**cpuX.l2cache-snoop\_data\_transfer\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-snoop\_issue\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.



**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**cpuX.semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**cpuX.semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**cpuX.semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**cpuX.semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**cpuX.semihosting-heap\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

**`dcache-maintenance_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

**`dcache-miss_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

**`dcache-prefetch_enabled`****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit

or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

### **`dcache-read_latency`**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

### **`dcache-size`**

#### **Type**

int

#### **Default value**

0x10000

#### **Description**

L1 D-Cache size in bytes.

### **`dcache-snoop_data_transfer_latency`**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

### **`dcache-state_modelled`**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Set whether D-cache has stateful implementation.

**dcache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**dcache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**default\_opmode****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**ecv\_support\_level****Type**

int

**Default value**

0x2

**Description**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CM0D****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CM0D value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int



**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x2

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**has\_rndr****Type**

int

**Default value**

0x1

**Description**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**has\_v8\_7\_spe\_inverted\_filtering****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

**has\_v8\_7\_spe\_previous\_branch\_target****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

**icache-read\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

**icache-size****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**icache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**instruction\_tlb\_size****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**invalidate\_code\_cache\_on\_icache\_cmo****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.

Note: Enabling this parameter will reduce simulation performance.

**l3cache-ways****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**log2\_trace\_buffer\_alignment****Type**

int

**Default value**

0x6

**Description**

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

**memory\_tagging\_support\_level****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpamidr\_has\_force\_ns****Type**

int

**Default value**

0x0

**Description**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpamidr\_has\_sdeflt****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpamidr\_has\_tidr****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_acp****Type**

int

**Default value**

0x0

**Description**

Number of ACP ports.



**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**rme\_level0\_gpt\_size****Type**

int

**Default value**

0x0

**Description**

The range of address space protected by each entry in the level 0 GPT (0->1GB 1->16GB, 2->64GB, 3->512GB).

**rme\_support\_level****Type**

int

**Default value**

0x2

**Description**

0 -> Realm management extension not implemented, 1 -> LEGACY\_TZ\_EN mode i.e. RME register fields are stateful but only supports secure/non-secure states, 2 -> Realm management extension fully implemented (FEAT\_RME).

**`rndr_rndrrs_seed`****Type**

int

**Default value**

0x0

**Description**

Initial seed for random engine used in RNDR register.

**`stage12_tlb_size`****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**`tcr_txsz_undersize_should_fault`****Type**

bool

**Default value**

0x0

**Description**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

**`tlb_latency`****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled**

Type  
bool

Default value  
0x0

Description  
If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat\_PAC\_as\_NOP**

Type  
bool

Default value  
0x0

Description  
Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency**

Type  
int

Default value  
0x0

Description  
Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

3.5.83 ARMNeoverseV3CT

ARMNeoverseV3CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-517: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About ARMNeoverseV3CT

A DSU-120 DynamIQ cluster containing a single Neoverse-V3 core configured for Direct connect.

The core supports the following optional features:

- Realm Management Extension
- Coherent instruction cache

The following DSU/CPU features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseV3CT

This model has the following Iris instances:

**Table 3-518: ARMNeoverseV3CT Iris instances**

InstanceName	ComponentName
ARMNeoverseV3CT	Cluster_ARM_Neoverse-V3
ARMNeoverseV3CT.AMU	PVBusLogger
ARMNeoverseV3CT.AMU.mapper	PVBusMapper
ARMNeoverseV3CT.DAP	PVBusLogger
ARMNeoverseV3CT.DAP.mapper	PVBusMapper
ARMNeoverseV3CT.DSU	DSU
ARMNeoverseV3CT.DSU.PPU_cluster	PPUv1
ARMNeoverseV3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV3CT.DSU.PPU_core0	PPUv1
ARMNeoverseV3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV3CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV3CT.DSU.shared_cache	PVCache
ARMNeoverseV3CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseV3CT.DSU.shared_cache.upstream[2]	PVBusSlave

InstanceName	ComponentName
ARMNeoverseV3CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseV3CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMNeoverseV3CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV3CT.MMAP	PVBusLogger
ARMNeoverseV3CT.MMAP.mapper	PVBusMapper
ARMNeoverseV3CT.RAS	PVBusLogger
ARMNeoverseV3CT.RAS.mapper	PVBusMapper
ARMNeoverseV3CT.cpu0	ARM_Neoverse-V3
ARMNeoverseV3CT.cpu0.UTLB	TLB
ARMNeoverseV3CT.cpu0.debug_rom	debug_rom
ARMNeoverseV3CT.cpu0.dtlb	TLB
ARMNeoverseV3CT.cpu0.l1dcache	PVCache
ARMNeoverseV3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.cpu0.l1icache	PVCache
ARMNeoverseV3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.cpu0.l2cache	PVCache
ARMNeoverseV3CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseV3CT.default_CTM	EmbeddedCT
ARMNeoverseV3CT.ext_bus	PVBusLogger
ARMNeoverseV3CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseV3CT.global_debug_rom	debug_rom
ARMNeoverseV3CT.secondary_debug_rom	debug_rom
ARMNeoverseV3CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

**Table 3-519: ARMNeoverseV3CT MTI instances**

InstanceName	ComponentName
ARMNeoverseV3CT.AMU	PVBusLogger
ARMNeoverseV3CT.AMU.mapper	PVBusMapper
ARMNeoverseV3CT.DAP	PVBusLogger
ARMNeoverseV3CT.DAP.mapper	PVBusMapper
ARMNeoverseV3CT.DSU	DSU
ARMNeoverseV3CT.DSU.PPU_cluster	PPUv1
ARMNeoverseV3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV3CT.DSU.PPU_core0	PPUv1
ARMNeoverseV3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV3CT.DSU.mpam_busslave	PVBusSlave

InstanceName	ComponentName
ARMNeoverseV3CT.DSU.shared_cache	PVCache
ARMNeoverseV3CT.DSU.shared_cache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.DSU.shared_cache.upstream[1]	PVBusSlave
ARMNeoverseV3CT.DSU.shared_cache.upstream[2]	PVBusSlave
ARMNeoverseV3CT.DSU.shared_cache.upstream[3]	PVBusSlave
ARMNeoverseV3CT.DSU.shared_cache.upstream[4]	PVBusSlave
ARMNeoverseV3CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV3CT.MMAP	PVBusLogger
ARMNeoverseV3CT.MMAP.mapper	PVBusMapper
ARMNeoverseV3CT.RAS	PVBusLogger
ARMNeoverseV3CT.RAS.mapper	PVBusMapper
ARMNeoverseV3CT.cpu0	ARM_Neoverse-V3
ARMNeoverseV3CT.cpu0.UTLB	TLB
ARMNeoverseV3CT.cpu0.l1dcache	PVCache
ARMNeoverseV3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.cpu0.l1icache	PVCache
ARMNeoverseV3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.cpu0.l2cache	PVCache
ARMNeoverseV3CT.cpu0.l2cache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.cpu0.l2cache.upstream[1]	PVBusSlave
ARMNeoverseV3CT.ext_bus	PVBusLogger
ARMNeoverseV3CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMNeoverseV3CT

**Table 3-520: Ports**

Name	Protocol	Type	Description
acp_s[2]	PVBus	Slave	AXI ACP slave port.
AEND0MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	Value_64	Slave	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).

Name	Protocol	Type	Description
ASTART2MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	Value_64	Slave	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	Signal	Slave	CHI defined pins.
broadcastcachemaint	Signal	Slave	ACE defined pins.
broadcastouter	Signal	Slave	ACE defined pins.
broadcastpersist	Signal	Slave	CHI defined pins.
cfgend[1]	Signal	Slave	This signal is for EE bit initialisation.
cfgsdisable	Signal	Slave	This signal disables write access to some secure Interrupt Controller registers.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsmp_channel	PChannel	Master	Cluster PCSM signal
cluster_powerdown_out	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
clusterid	Value	Slave	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.
clusterpmuirq	Signal	Master	DynamlQ pmu irq
CNTHPIRQ[1]	Signal	Master	Timer signals to SOC.
CNTHPSIRQ[1]	Signal	Master	Timer signals to SOC
CNTHVIRQ[1]	Signal	Master	Timer signals to SOC.
CNTHVSIRQ[1]	Signal	Master	Timer signals to SOC
CNTPNSIRQ[1]	Signal	Master	Timer signals to SOC.
CNTPSIRQ[1]	Signal	Master	Timer signals to SOC.
cntvalueb	CounterInterface	Slave	Interface to SoC level counter module.
CNTVIRQ[1]	Signal	Master	Timer signals to SOC.
commirq[1]	Signal	Master	Interrupt signal from debug communications channel.
core_clk_in[1]	ClockSignal	Slave	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.

Name	Protocol	Type	Description
core_pcsmpchannel[1]	PChannel	Master	Core PCSM signals
core_powerdown_out[1]	Signal	Master	DEPRECATED - An output from PPU that informs thermal controller of the core power info
coreerrirq[1]	Signal	Master	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq[1]	Signal	Master	Fault indicator for a detected 1 or 2 bit ECC error
cryptodisable[1]	Signal	Slave	Disable cryptography extensions after reset.
cti[1]	v8EmbeddedCrossTrigger_controlprotocol	Master	Cross trigger matrix port.
ctidbgirq[1]	Signal	Master	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	Signal	Slave	External debug interface.
dbgnopwrdown[1]	Signal	Master	No power-down request.
dbgpwrupreq[1]	Signal	Master	Debug power up request.
defaultMP	Signal	Slave	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	PVBus	Slave	External debug interface.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE.
fiq[1]	Signal	Slave	This signal drives the CPUs fast-interrupt handling.
gicreset	Signal	Master	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s[1]	GICv3Comms	Slave	GICv3 AXI-stream port.
irq[1]	Signal	Slave	This signal drives the CPUs interrupt handling.
l0gptsz	Value	Slave	RME LOGPTSZ port
legacy_tz_en	Signal	Slave	RME LEGACY_TZ_EN port
memorymapped_debug_s	PVBus	Slave	External debug interface.
pmbirq[1]	Signal	Master	Interrupt signal from the statistical profiling unit.
pmuirq[1]	Signal	Master	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	Signal	Master	PPU cluster interrupt.
ppu_cluster_wakerequest	Signal	Slave	PPU cluster wake request signal.
ppu_core_irq[1]	Signal	Master	PPU core interrupt.
ppu_core_wakerequest[1]	Signal	Slave	PPU core wake request signal.



Name	Protocol	Type	Description
presetdbg	Signal	Slave	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	PVBus	Master	The core will generate bus requests on this port.
pvbus_periph_m	PVBus	Master	The core can generate peripheral bus request on this port.
rei[1]	Signal	Slave	Per core RAM Error Interrupt.
reset	Signal	Slave	Reset signal to cluster.
rlpiden	Signal	Slave	External debug interface.
rtpiden	Signal	Slave	External debug interface.
rvbaraddr[1]	Value_64	Slave	Reset vector base address.
sci_m	SystemCoherencyInterface	Master	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei[1]	Signal	Slave	Per core System Error physical pins.
spiden	Signal	Slave	External debug interface.
ticks[1]	InstructionCount	Master	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq[1]	Signal	Master	Interrupt signal from the trace buffer unit.
utility_bus_s	PVBus	Slave	Utility bus slave
vcpumntirq[1]	Signal	Master	Interrupt signal for virtual CPU maintenance IRQ.
vfiq[1]	Signal	Slave	Virtualised FIQ.
virq[1]	Signal	Slave	Virtualised IRQ.
virtio_s	PVBus	Slave	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei[1]	Signal	Slave	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseV3CT

### AEND0\_DEFAULT

#### Type

int

#### Default value

0x0

#### Description

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

**AEND1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

**AEND2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

**AEND3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

**ASTART0\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

**ASTART1\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

**ASTART2\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

**ASTART3\_DEFAULT****Type**

int

**Default value**

0x0

**Description**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

**BROADCASTATOMIC****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

**BROADCASTCACHEMAINT****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

**BROADCASTOUTER****Type**

bool

**Default value**

0x0

**Description**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

**BROADCASTPERSIST****Type**

bool

**Default value**

0x1

**Description**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

**CLUSTER\_ID****Type**

int

**Default value**

0x0

**Description**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the MasterID and ManagerID64 of the core.

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled****Type**

int

**Default value**

0x1

**Description**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. Changing this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain CMOs even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required .

**CPUCFR****Type**

int

**Default value**

0x20

**Description**

Value of CPU Configuration Register.

**GICDISABLE****Type**

bool

**Default value**

0x1

**Description**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

**NUM\_CORES****Type**

int

**Default value**

0x1

**Description**

Number of cores per cluster.

**brbe\_log2\_num\_records****Type**

int

**Default value**

0x5

**Description**

Log2 of number of BRB records supported. 3 -> 8 records, ... 6 -> 64 records.

**bus\_type****Type**

int

**Default value**

0x0

**Description**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

**core\_power\_on\_by\_default****Type**

bool

**Default value**

0x0

**Description**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

Divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

Multiplier for calculating CPI (Cycles Per Instruction).

**cpuX.CFGEND****Type**

bool

**Default value**

0x0

**Description**

Endianness configuration at reset. 0, little endian. 1, big endian.

**cpuX.CFGTE****Type**

bool

**Default value**

0x0

**Description**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

**cpuX.CRYPTODISABLE****Type**

bool

**Default value**

0x0

**Description**

Disable cryptographic features.

**cpuX.RVBARADDR****Type**

int

**Default value**

0x0

**Description**

Value of RVBAR\_ELx register.

**cpuX.crypto\_aes****Type**

int

**Default value**

0x2

**Description**

AES instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 2, AES and PMULL instructions implemented (FEAT\_AES, FEAT\_PMULL).

**cpuX.crypto\_sha3****Type**

int

**Default value**

0x2

**Description**

Implement ARMv8.4 SHA-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA3). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.crypto\_sha512****Type**

int

**Default value**

0x2

**Description**

Implement ARMv8.4 SHA-512 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA512). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.crypto\_sm3****Type**

int

**Default value**

0x2

**Description**

Implement ARMv8.4 SM-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM3). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.crypto\_sm4****Type**

int

**Default value**

0x2

**Description**

Implement ARMv8.4 SM-4 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM4). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.4 is enabled. - 2, feature is implemented.

**cpuX.enable\_trace\_special\_hlt\_imm16****Type**

bool



**Default value**

0x0

**Description**

Enable usage of parameter trace\_special\_hlt\_imm16.

**cpuX.l2cache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-read\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-

read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

### **cpuX.l2cache-read\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

### **cpuX.l2cache-size**

#### **Type**

int

#### **Default value**

0x200000

#### **Description**

L2 Cache size in bytes.

### **cpuX.l2cache-snoop\_data\_transfer\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

### **cpuX.l2cache-snoop\_issue\_latency**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-ways****Type**

int

**Default value**

0x8

**Description**

L2 Cache number of ways (sets are implicit from size).

**cpuX.l2cache-write\_access\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

**cpuX.l2cache-write\_latency****Type**

int

**Default value**

0x0

**Description**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

**cpuX.max\_code\_cache\_mb****Type**

int

**Default value**

0x100

**Description**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

**cpuX.min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**cpuX.semihosting-A32\_HLT****Type**

int

**Default value**

0xf000

**Description**

A32 HLT number for semihosting calls.

**cpuX.semihosting-A64\_HLT****Type**

int

**Default value**

0xf000

**Description**

A64 HLT number for semihosting calls.

**cpuX.semihosting-ARM\_SVC****Type**

int

**Default value**

0x123456

**Description**

A32 SVC number for semihosting calls.

**cpuX.semihosting-T32\_HLT****Type**

int

**Default value**

0x3c

**Description**

T32 HLT number for semihosting calls.

**cpuX.semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting calls.

**`cpuX.semihosting-cmd_line`****Type**

string

**Default value**

""

**Description**

Command line available to semihosting calls.

**`cpuX.semihosting-cwd`****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**`cpuX.semihosting-enable`****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC/HLT traps.

**`cpuX.semihosting-heap_base`****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**`cpuX.semihosting-heap_limit`****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of top of heap.

**cpuX.semihosting-stack\_base****Type**

int

**Default value**

0x10000000

**Description**

Virtual address of base of descending stack.

**cpuX.semihosting-stack\_limit****Type**

int

**Default value**

0xf000000

**Description**

Virtual address of stack limit.

**cpuX.trace\_special\_hlt\_imm16****Type**

int

**Default value**

0xf000

**Description**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

**cpuX.vfp-enable\_at\_reset****Type**

bool

**Default value**

0x0

**Description**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

**cpuX.vfp-present****Type**

bool

**Default value**

0x1

**Description**

Set whether the model has VFP support.

**dcache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

**dcache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

**dcache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

**dcache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

**`dcache-read_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

**`dcache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

**`dcache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 D-Cache size in bytes.

**`dcache-snoop_data_transfer_latency`****Type**

int



**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

**`dcache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether D-cache has stateful implementation.

**`dcache-write_access_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

**`dcache-write_latency`****Type**

int

**Default value**

0x0

**Description**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

**`default_opmode`****Type**

int

**Default value**

0x4

**Description**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

**diagnostics****Type**

bool

**Default value**

0x0

**Description**

Enable DynamIQ diagnostic messages.

**ecv\_support\_level****Type**

int

**Default value**

0x2

**Description**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

**enable\_simulation\_performance\_optimizations****Type**

bool

**Default value**

0x1

**Description**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

**ete.CLAIMTAGS****Type**

int

**Default value**

0x4

**Description**

Number of claim tags.

**ete.MAX\_INST\_PER\_Q****Type**

int

**Default value**

0x1

**Description**

Maximum limit for the number of instructions implied by a Q element.

**ete.NumberOfRSPairs****Type**

int

**Default value**

0x8

**Description**

Number of resource selector pairs.

**ete.PIDR\_CMOD****Type**

int

**Default value**

0x0

**Description**

TRCPIDR CMOD value.

**ete.Q\_CADENCE****Type**

int

**Default value**

0x1

**Description**

Number of instruction blocks traced between two Q elements.

**ete.RES0\_STATEFUL****Type**

bool

**Default value**

0x0

**Description**

Whether RES0 bits are stateful or RAZ/WI.

**ete.RETSTACK****Type**

int

**Default value**

0x3

**Description**

Return stack depth.

**ete.SIM\_OVERFLOW\_GRANULARITY****Type**

int

**Default value**

0x64

**Description**

Number of instruction blocks in each granule, for simulated overflow.

**ete.SIM\_OVERFLOW\_PERCENTAGE****Type**

int

**Default value**

0x0

**Description**

Percentage of instruction blocks lost in each granule, for simulated overflow.

**ete.SOURCE\_ADDRESS****Type**

bool

**Default value**

0x0

**Description**

Allow generation of source address elements.

**ete.TRACE\_OUTPUT****Type**

string

**Default value**

""

**Description**

File to which to write trace byte stream.

**ete.TRCRSRTA\_FORCED\_EXCEP****Type**

bool

**Default value**

0x0

**Description**

TRCRSR.TA value for a forcibly traced exception.

**ext\_abort\_so\_write\_ras\_type****Type**

int

**Default value**

0x2

**Description**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks****Type**

bool

**Default value**

0x0

**Description**

Force MTE tag accesses to RAZ/WI and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

**force\_zero\_PSTATE\_PAN****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

**force\_zero\_mpam\_partid\_and\_pmg****Type**

int

**Default value**

0x0

**Description**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect.  
1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

**has\_coherent\_icache****Type**

bool

**Default value**

0x1

**Description**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

**has\_enhanced\_pan****Type**

int

**Default value**

0x2

**Description**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) Possible values of this parameter are: - 1, feature is implemented if ARMv8.7 is enabled. - 2, feature is implemented.

**has\_ete****Type**

bool

**Default value**

0x0

**Description**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (--plugin or -P).

**has\_peripheral\_port****Type**

bool

**Default value**

0x0

**Description**

If true, an additional AXI peripheral port is configured.

**has\_rndr****Type**

int

**Default value**

0x1

**Description**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG). Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

**has\_statistical\_profiling****Type**

bool

**Default value**

0x1

**Description**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

**has\_v8\_7\_spe\_inverted\_filtering****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

**has\_v8\_7\_spe\_previous\_branch\_target****Type**

bool

**Default value**

0x0

**Description**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

**icache-hit\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

**icache-maintenance\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

**icache-miss\_latency****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

**icache-prefetch\_enabled****Type**

bool

**Default value**

0x0

**Description**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

**icache-read\_access\_latency****Type**

int



**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

**`icache-read_latency`****Type**

int

**Default value**

0x0

**Description**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

**`icache-size`****Type**

int

**Default value**

0x10000

**Description**

L1 I-Cache size in bytes.

**`icache-state_modelled`****Type**

bool

**Default value**

0x0

**Description**

Set whether I-cache has stateful implementation.

**`instruction_tlb_size`****Type**

int

**Default value**

0x0

**Description**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

**`invalidate_code_cache_on_icache_cmo`****Type**

int

**Default value**

0x0

**Description**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE.  
Note: Enabling this parameter will reduce simulation performance.

**`l3cache-ways`****Type**

int

**Default value**

0x10

**Description**

L3 Cache number of ways (sets are implicit from size).

**`log2_trace_buffer_alignment`****Type**

int

**Default value**

0x6

**Description**

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

**`memory_tagging_support_level`****Type**

int

**Default value**

0x3

**Description**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

**mpamidr\_has\_force\_ns****Type**

int

**Default value**

0x0

**Description**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpamidr\_has\_sdeflt****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpamidr\_has\_tidr****Type**

int

**Default value**

0x1

**Description**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear. Possible values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.6 is enabled. - 2, feature is implemented.

**mpmm\_accumulator\_multiplier****Type**

int

**Default value**

0x1

**Description**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. A value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). This is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

**num\_acp****Type**

int

**Default value**

0x0

**Description**

Number of ACP ports.

**ptw\_latency****Type**

int

**Default value**

0x0

**Description**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**rme\_level0\_gpt\_size****Type**

int

**Default value**

0x0

**Description**

The range of address space protected by each entry in the level 0 GPT (0->1GB 1->16GB, 2->64GB, 3->512GB).

**rme\_support\_level****Type**

int

**Default value**

0x2

**Description**

0 -> Realm management extension not implemented, 1 -> LEGACY\_TZ\_EN mode i.e. RME register fields are stateful but only supports secure/non-secure states, 2 -> Realm management extension fully implemented (FEAT\_RME).

**rndr\_rndrrs\_seed****Type**

int

**Default value**

0x0

**Description**

Initial seed for random engine used in RNDR register.

**stage12\_tlb\_size****Type**

int

**Default value**

0x80

**Description**

Number of stage1+2 tlb entries.

**tcr\_txsz\_undersize\_should\_fault****Type**

bool

**Default value**

0x0

**Description**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

**tlb\_latency****Type**

int

**Default value**

0x0

**Description**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

**tlbi\_stall\_enabled****Type**

bool

**Default value**

0x0

**Description**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

**treat\_PAC\_as\_NOP****Type**

bool

**Default value**

0x0

**Description**

Non-architecture parameter. Treat Pointer Authentication as NOP. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are NOP. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

**walk\_cache\_latency****Type**

int

**Default value**

0x0

**Description**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

### 3.5.84 ARMSC000CT

ARMSC000CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-521: IP revisions support**

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Differences between the model and the RTL

The model has the following limitations:

- It does not implement any security features.
- Only bit[0] of the Auxiliary Control Register is supported for read/write. No functionality is implemented.
- The Security Features Control Register read/write access is supported using `SECKEY`. No functionality is implemented.

#### Iris and MTI instances for ARMSC000CT

This model has the following Iris instances:

**Table 3-522: ARMSC000CT Iris instances**

InstanceName	ComponentName
ARMSC000CT	ARM_SC000
ARMSC000CT.acp_mapper	PVBusMapper
ARMSC000CT.ext_bus	PVBusLogger
ARMSC000CT.ext_bus.mapper	PVBusMapper
ARMSC000CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-523: ARMSC000CT MTI instances**

InstanceName	ComponentName
ARMSC000CT	ARM_SC000
ARMSC000CT.acp_mapper	PVBusMapper
ARMSC000CT.ext_bus	PVBusLogger
ARMSC000CT.ext_bus.mapper	PVBusMapper
ARMSC000CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMSC000CT

Table 3-524: Ports

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
currpri	Value	Master	Current execution priority.
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr[32]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
io_port_in	PVBus	Slave	I/O port pair. See the documentation for the io_port_out port.
io_port_out	PVBus	Master	I/O port pair. Used if IOP is true. Transactions from io_port_out which do not "match" should be returned via io_port_in. For performance reasons, the I/O port interface is not modelled directly. Instead, a simple PVBus gasket is inserted at the point in the memory system where the I/O port would be. In hardware, a device would be attached to the port which would tell the CPU whether it would like to intercept each transaction, given its address. This can be modelled in a performant manner by connecting a PVBusMapper-based device to io_port_out which intercepts transactions of interest and passes other transactions back to the CPU via io_port_in. Your I/O port device model is also responsible for aborting transactions which would be aborted on hardware (e.g. exclusives) if necessary.
lockup	Signal	Master	Asserted when the processor is in lockup state.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
stcalib	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARMSC000CT

**BIGENDINIT****Type**

bool

**Default value**

0x0



**Description**

Initialize processor to big endian mode.

**BKPT****Type**

int

**Default value**

0x4

**Description**

Number of breakpoint unit comparators implemented.

**DBG****Type**

bool

**Default value**

0x1

**Description**

Set whether debug extensions are implemented.

**IOP****Type**

bool

**Default value**

0x0

**Description**

Send all d-side transactions to the port, io\_port\_out. Transactions which do not match should be returned to the port, io\_port\_in.

**IRQDIS****Type**

int

**Default value**

0x0

**Description**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n].

**NUM\_IRQ****Type**

int

**Default value**

0x20

**Description**

Number of user interrupts.

**NUM\_MPU\_REGION****Type**

int

**Default value**

0x0

**Description**

Number of MPU regions.

**SYST****Type**

bool

**Default value**

0x1

**Description**

Enable support for SysTick timer functionality.

**USER****Type**

bool

**Default value**

0x1

**Description**

Enable support for Unprivileged/Privileged Extension.

**VTOR****Type**

bool

**Default value**

0x1

**Description**

Include Vector Table Offset Register.

**WIC****Type**

bool

**Default value**

0x1

**Description**

Include support for WIC-mode deep sleep.

**WPT****Type**

int

**Default value**

0x2

**Description**

Number of watchpoint unit comparators implemented.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**master\_id****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**semihosting-enable****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**semihosting-heap\_base****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**semihosting-heap\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**semihosting-stack\_base****Type**

int

**Default value**

0x20800000

**Description**

Virtual address of base of descending stack.

**semihosting-stack\_limit****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of stack limit.

### 3.5.85 ARMSC300CT

ARMSC300CT CPU component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-525: IP revisions support**

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Differences between the model and the RTL**

The model has the following limitations:

- It does not implement any security features.
- The Trash Register is implemented as **RAZ/WI**.
- Only bit[0] of the Auxiliary Control Register is supported for read/write. No functionality is implemented.
- The Security Features Control Register read/write access is supported using **SECKEY**. No functionality is implemented.

**Implementation of ITM in M-class models**

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an **ITM\_PACKET\_TYPE** field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.

Field value	Description	Supported by model
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

## Iris and MTI instances for ARMSC300CT

This model has the following Iris instances:

**Table 3-527: ARMSC300CT Iris instances**

InstanceName	ComponentName
ARMSC300CT	ARM_SC300
ARMSC300CT.acp_mapper	PVBusMapper
ARMSC300CT.ext_bus	PVBusLogger
ARMSC300CT.ext_bus.mapper	PVBusMapper
ARMSC300CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

**Table 3-528: ARMSC300CT MTI instances**

InstanceName	ComponentName
ARMSC300CT	ARM_SC300
ARMSC300CT.acp_mapper	PVBusMapper
ARMSC300CT.ext_bus	PVBusLogger
ARMSC300CT.ext_bus.mapper	PVBusMapper
ARMSC300CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMSC300CT

**Table 3-529: Ports**

Name	Protocol	Type	Description
ahbd	PVBus	Slave	Debug AHB - core bus slave driven by the DAP.
auxfault	Value	Slave	This is wired to the Auxiliary Fault Status Register.
bigend	Signal	Slave	Configure big endian data format.
clk_in	ClockSignal	Slave	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
currpri	Value	Master	Current execution priority.

Name	Protocol	Type	Description
edbgrq	Signal	Slave	External debug request.
event	Signal	Peer	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr[240]	Signal	Slave	This signal array delivers signals to the NVIC.
intnmi	Signal	Slave	Configure non maskable interrupt.
lockup	Signal	Master	Asserted when the processor is in lockup state.
poreset	Signal	Slave	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	PVBus	Master	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	PVBus	Master	The core will generate bus requests on this port.
sleepdeep	Signal	Master	Asserted when the processor is in deep sleep.
sleeping	Signal	Master	Asserted when the processor is in sleep.
stcalib	Value	Slave	This is the calibration value for the SysTick timer.
stclk	ClockSignal	Slave	This is the reference clock for the SysTick timer.
sysreset	Signal	Slave	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	Signal	Master	Asserted to indicate that a reset is required.
ticks	InstructionCount	Master	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARMSC300CT

### BB\_PRESENT

#### Type

bool

#### Default value

0x1

#### Description

Enable bitbanding.

### BIGENDINIT

#### Type

bool

#### Default value

0x0

#### Description

Initialize processor to big endian mode.

### DBGLVL

#### Type

int



**Default value**

0x3

**Description**

0: No debug; 1: Minimal debug; 2: Full debug without DWT address matching; 3: Full debug support with DWT data-comparators.

**LVL\_WIDTH****Type**

int

**Default value**

0x3

**Description**

Number of bits of interrupt priority.

**NUM\_IRQ****Type**

int

**Default value**

0x10

**Description**

Number of user interrupts.

**NUM\_MPU\_REGION****Type**

int

**Default value**

0x8

**Description**

Number of MPU regions.

**TRACE\_LVL****Type**

int

**Default value**

0x1

**Description**

Level of instrumentation trace supported. 0: No trace, 1: Standard trace. ITM, TPIU and DWT triggers and counters present, 2: Full trace. ITM, TPIU, ETM and DWT triggers and counters present.

**WIC****Type**

bool

**Default value**

0x1

**Description**

Include support for WIC-mode deep sleep.

**cpi\_div****Type**

int

**Default value**

0x1

**Description**

divider for calculating CPI (Cycles Per Instruction).

**cpi\_mul****Type**

int

**Default value**

0x1

**Description**

multiplier for calculating CPI (Cycles Per Instruction).

**master\_id****Type**

int

**Default value**

0x0

**Description**

Master ID presented in bus transactions.

**min\_sync\_level****Type**

int

**Default value**

0x0

**Description**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

**reported\_patch\_level****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**reported\_revision\_number****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

**semihosting-Thumb\_SVC****Type**

int

**Default value**

0xab

**Description**

T32 SVC number for semihosting.

**semihosting-cmd\_line****Type**

string

**Default value**

""

**Description**

Command line available to semihosting SVC calls.

**semihosting-cwd****Type**

string

**Default value**

""

**Description**

Base directory for semihosting file access.

**`semihosting-enable`****Type**

bool

**Default value**

0x1

**Description**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

**`semihosting-heap_base`****Type**

int

**Default value**

0x0

**Description**

Virtual address of heap base.

**`semihosting-heap_limit`****Type**

int

**Default value**

0x20700000

**Description**

Virtual address of top of heap.

**`semihosting-stack_base`****Type**

int

**Default value**

0x20800000

**Description**

Virtual address of base of descending stack.

**semihosting-stack\_limit**

**Type**  
int

**Default value**  
0x20700000

**Description**  
Virtual address of stack limit.

3.6 Media components

This section describes the Media components.

3.6.1 D71

ARM D71 Display Processor. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-530: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Changes in 11.29.19

Ports added:

- display\_trace

About D71

The model has the following limitations:

- No support for trusted layers.
- No support for image enhancements.
- No coprocessor support for HDR processing.
- No QoS support.
- The following configuration parameters are not available:
  - CONFIG\_MAX\_LINE\_SIZE
  - CONFIG\_DISPLAY\_TBU\_EN. TBUs are integrated separately using the given ports.
  - CONFIG\_AFBC\_DMA\_EN. The ADU is present. If it is not used, do not program it.

- Some image formats are unsupported. For details, see the next section, Supported image formats.

## Supported image formats

### ADU DS\_FORMAT

The `DS_FORMAT` register defines the image formats supported by the ADU DMA subsystem.

Support for the following image formats is implemented:

- All image formats supported.

### ADU AES\_FORMAT

The `AES_FORMAT` register defines the image formats supported by the ADU AFBC encoding subsystem.

Support for the following image formats is implemented:

- RGB\_888
- RGBA\_8888
- YUV\_420\_P2\_8

### LS\_FORMAT/LR\_FORMAT

The `LS_FORMAT` and `LR_FORMAT` registers define the image formats supported by the main pipeline's layer processing unit.

Support for uncompressed images in the following formats is implemented:

- ARGB\_2101010
- BGRA\_1010102
- ARGB\_8888
- ABGR\_8888
- RGBA\_8888
- BGRA\_8888
- XRGB\_8888
- XBGR\_8888
- RGBX\_8888
- BGRX\_8888
- RGB\_888
- BGR\_888
- RGBA\_5551
- ABGR\_1555
- RGB\_565
- BGR\_565

- VYUY\_422\_P1\_8
- YVYU\_422\_P1\_8
- YUV\_420\_P2\_8
- YUV\_420\_P3\_8
- YUV\_420\_P1\_10
- YUV\_420\_P2\_10

Support for compressed images in the following formats is implemented:

- ABGR\_2101010
- ABGR\_8888
- BGR\_888
- ABGR\_1555
- BGR\_565
- YUV\_422\_P2\_8
- YUV\_420\_P2\_8
- YUV\_420\_P2\_10

## LW\_FORMAT

The `LW_FORMAT` register defines the image formats supported by the memory-writeback scheme performed by the layer processing unit.

Support for the following image formats is implemented:

- ARGB\_2101010
- ABGR\_2101010
- RGBA\_1010102
- BGRA\_1010102
- ARGB\_8888
- ABGR\_8888
- RGBA\_8888
- BGRA\_8888
- XRGB\_8888
- XBGR\_8888
- RGBX\_8888
- BGRX\_8888
- RGB\_888
- BGR\_888
- YUV\_420\_P2\_8

## Iris and MTI instances for D71

This model has the following Iris instances:

**Table 3-531: D71 Iris instances**

InstanceName	ComponentName
D71	D71
D71.apb_slave_adu	PVBusSlave
D71.apb_slave_dpu	PVBusSlave

This model has the following MTI trace components:

**Table 3-532: D71 MTI instances**

InstanceName	ComponentName
D71	D71
D71.apb_slave_adu	PVBusSlave
D71.apb_slave_dpu	PVBusSlave

## Ports for D71

**Table 3-533: Ports**

Name	Protocol	Type	Description
apb_pvbus_s_adu	PVBus	Slave	Slave port for register access.
apb_pvbus_s_dpu	PVBus	Slave	-
axi_pvbus_adu_m	PVBus	Master	Master AXI port for the AFBC unit
axi_pvbus_lpu_m[2]	PVBus	Master	Master AXI ports for pipelines
display[2]	LCD	Master	LCD ports for display outputs
display_trace[2]	FrameTracingProtocol	Master	FrameTrace port
irq0_gcu_out	Signal	Master	Shared interrupt owned by the GCU
irq1_adu_out	Signal	Master	Interrupt signal for the ADU block
pixelclock_in[2]	ClockSignal	Slave	Pixel clock inputs for the display outputs
pvbus_tbu_m[2]	PVBus	Master	Master ports for connection to TBU (SMMUv3)
pvbus_tbu_s[2]	PVBus	Slave	Slave ports for loopback from TBU (SMMUv3)
reset_signal	Signal	Slave	Reset signal.

## Parameters for D71

### adu\_nprot\_nsaid

#### Type

int

#### Default value

0x0



**Description**

Non-protected NSAID for ADU transactions.

**adu\_nprot\_s2\_sid**

**Type**

int

**Default value**

0x2

**Description**

Stage 2 non-protected StreamID for ADU transactions.

**adu\_prot\_nsaaid**

**Type**

int

**Default value**

0x1

**Description**

Protected NSAID for ADU transactions.

**adu\_prot\_s2\_sid**

**Type**

int

**Default value**

0x5

**Description**

Stage 2 protected StreamID for ADU transactions.

**adu\_rd\_s1\_sid**

**Type**

int

**Default value**

0xa

**Description**

Stage 1 StreamID for ADU DMA read layer.

**adu\_wr\_s1\_sid**

**Type**

int

**Default value**

0xb

**Description**

Stage 1 StreamID for ADU AES write-back layer.

**display\_split\_en****Type**

int

**Default value**

0x0

**Description**

Display split enabled or not.

**force\_frame\_rate\_0****Type**

int

**Default value**

0x0

**Description**

If 0 PXLCLK0 is used, if >0 the model refreshes display output 0 at the rate per simulated second.

**force\_frame\_rate\_1****Type**

int

**Default value**

0x0

**Description**

If 0 PXLCLK1 is used, if >0 the model refreshes display output 1 at the rate per simulated second.

**lpu0\_l0\_s1\_sid****Type**

int

**Default value**

0x0

**Description**

Stage 1 StreamID for LPU0 read layer 0.

**lpu0\_l1\_s1\_sid****Type**

int

**Default value**

0x1

**Description**

Stage 1 StreamID for LPU0 read layer 1.

**lpu0\_l2\_s1\_sid****Type**

int

**Default value**

0x2

**Description**

Stage 1 StreamID for LPU0 read layer 2.

**lpu0\_l3\_s1\_sid****Type**

int

**Default value**

0x3

**Description**

Stage 1 StreamID for LPU0 read layer 3.

**lpu0\_nprot\_nsaaid****Type**

int

**Default value**

0x0

**Description**

Non-protected NSAID for LPU0 transactions.

**lpu0\_nprot\_s2\_sid****Type**

int

**Default value**

0x0

**Description**

Stage 2 non-protected StreamID for LPU0 transactions.

**lpu0\_prot\_nsaaid****Type**

int

**Default value**

0x1

**Description**

Protected NSAID for LPU0 transactions.

**lpu0\_prot\_s2\_sid****Type**

int

**Default value**

0x3

**Description**

Stage 2 protected StreamID for LPU0 transactions.

**lpu0\_wr\_s1\_sid****Type**

int

**Default value**

0x8

**Description**

Stage 1 StreamID for LPU0 write-back layer.

**lpu1\_l0\_s1\_sid****Type**

int

**Default value**

0x4

**Description**

Stage 1 StreamID for LPU1 read layer 0.

**lpu1\_l1\_s1\_sid****Type**

int

**Default value**

0x5

**Description**

Stage 1 StreamID for LPU1 read layer 1.

**lpu1\_l2\_s1\_sid****Type**

int

**Default value**

0x6

**Description**

Stage 1 StreamID for LPU1 read layer 2.

**lpu1\_l3\_s1\_sid****Type**

int

**Default value**

0x7

**Description**

Stage 1 StreamID for LPU1 read layer 3.

**lpu1\_nprot\_nsaid****Type**

int

**Default value**

0x0

**Description**

Non-protected NSAID for LPU1 transactions.

**lpu1\_nprot\_s2\_sid****Type**

int

**Default value**

0x1

**Description**

Stage 2 non-protected StreamID for LPU1 transactions.

**lpu1\_prot\_nsaid****Type**

int

**Default value**

0x1

**Description**

Protected NSAID for LPU1 transactions.

**lpu1\_prot\_s2\_sid****Type**

int

**Default value**

0x4

**Description**

Stage 2 protected StreamID for LPU1 transactions.

**lpul\_wr\_sl\_sid**

**Type**

int

**Default value**

0x9

**Description**

Stage 1 StreamID for LPU1 write-back layer.

**num\_rich\_layers**

**Type**

int

**Default value**

0x2

**Description**

Number of Rich layers in each Layer Processing Unit.

3.6.2 FrameTracingComponent

Frame Tracing Component. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-534: IP revisions support

Revision	Quality level
r0p0	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About FrameTracingComponent

You can add the FrameTracingComponent to a platform to intercept frame buffers on the way to the display, and perform an action based on the content of the frame buffer. This is expected to be used to test components in the display path, for example GPUs, ISPs, displays, and video.

Connect the FrameTracingComponent to an HDLCD or D71 component using its `frame_trace_s` port to receive copies of displayed frames through `FrameTracingProtocol`.

## Configuration

Configure the behavior of the `FrameTracingComponent` using a JSON input file. The configuration file defines a list of frames and a list of actions.

A frame is simply a string identifier and a path to a binary file containing the frame buffer content. Actions refer to frames by using the string identifier.

An action inspects frames passed using `FrameTracingProtocol` and calls specific procedures based on the content of these frames.

Each action takes different arguments. The `FrameTracingComponent` supports the following actions:

**Table 3-535: FrameTracingComponent actions**

Action	Description
EXIT	End the simulation on matching a trigger frame.
TIME	Record the wall clock time before a trigger frame or between two frames.
RECORD	Save frames to files for offline processing.

You can define an arbitrary number of frames and actions.

The following JSON contains an example defining some frames, and one of each action:

```
{
  "frames" : [
    {
      "id" : "idstring1",
      "path" : "filepath1"
    },
    {
      "id" : "idstringN",
      "path" : "filepathN"
    },
    <...>
  ],
  "actions" : [
    {
      "action" : "EXIT",
      "frame" : "idstring1",
      "delay" : N
    },
    {
      "action" : "TIME",
      "id" : "actionid",
      "start_frame" : "idstring2",
      "stop_frame" : "idstring3",
    },
    {
      "action" : "RECORD",
      "start_frame" : "idstring2",
      "stop_frame" : "idstring4",
      "prefix" : "output_file_prefix",
      "start_delay" : N,
      "stop_delay" : M
    },
    <...>
  ]
}
```

There are 6 types of parameter used in the JSON configuration:

#### Frame identifier

An arbitrary string that uniquely identifies a particular frame. In the `frames` array, the `id` parameter defines the identifier. Actions only refer to frame identifiers.

#### Action type

The `action` parameter in the `actions` array. Must contain a valid action type. These parameters are case-insensitive.

#### Action identifier

Identifier to distinguish multiple actions of the same type in the log file.

#### File path

May be absolute or relative. Relative paths are relative to the directory containing the configuration file.

#### Output file prefix

May include path components, but the directory must exist. Relative paths are relative to the directory containing the configuration file. A sequential index is appended to this string when saving frame buffers.

#### Integer frame count

A delay applied to an aspect of an action by a number of frames. When not set, these default to 0, meaning no delay.

When the `ignore_consecutive_duplicates` parameter is true, a duplicate frame is considered to be the same as the previous frame and is not considered for any actions.

**Table 3-536: Action types and parameters**

Action	Parameter	Type	Description
EXIT	<code>frame</code>	Frame identifier	On matching <code>frame</code> , exit the simulation after <code>delay</code> frames.
	<code>delay</code>	Frame count	If <code>frame</code> is not specified, exit after <code>delay</code> frames from the start of the simulation.
TIME	<code>start_frame</code>	Frame identifier	Record the wall clock time between seeing <code>start_frame</code> and <code>stop_frame</code> .
	<code>stop_frame</code>	Frame identifier	If <code>start_frame</code> is not specified, record from the first input frame until seeing <code>stop_frame</code> .
	<code>id</code>	Action identifier	Recorded time will be printed in the log along with the string passed via <code>id</code> parameter.



Action	Parameter	Type	Description
RECORD	start_frame	Frame identifier	Record frames to files for offline processing.
	stop_frame	Frame identifier	The recording starts <code>start_delay</code> frames after seeing <code>start_frame</code> .
	prefix	Output file prefix	If <code>start_frame</code> is not specified, recording starts <code>start_delay</code> frames after the start of the simulation.
	start_delay	Frame count	The recording terminates on receiving <code>stop_delay</code> frames after seeing <code>stop_frame</code> .
	stop_delay	Frame count	If <code>stop_frame</code> is not specified, the recording terminates <code>stop_delay</code> frames after the start of the recording.
			If <code>start_delay</code> or <code>stop_delay</code> are not specified, they are assumed to be 0.  Frames are saved to files with the common prefix <code>prefix</code> , and an incrementing identifier suffix.

## Limitations

The FrameTracingComponent does not try to handle frame buffer formats. It assumes that width, height, and bits-per-pixel are sufficient to describe the size of the expected buffer, and compares the full content of the buffers.

## Iris and MTI instances for FrameTracingComponent

This model has the following Iris instances:

**Table 3-537: FrameTracingComponent Iris instances**

InstanceName	ComponentName
FrameTracingComponent	FrameTracingComponent

## Ports for FrameTracingComponent

**Table 3-538: Ports**

Name	Protocol	Type	Description
frame_trace_s	FrameTracingProtocol	Slave	Connect to `frame_trace_m`, the manager FrameTracingProtocol port of the source component to receive frame buffers as they are produced.

## Parameters for FrameTracingComponent

### config\_path

#### Type

string

#### Default value

""

Path to a JSON file containing a description of what the FrameTracingComponent should do.

**enabled**

**Type**  
bool

**Default value**  
true

Enables the frame tracing component. When disabled, the config file is not read, and the component is inactive.

**ignore\_consecutive\_duplicates**

**Type**  
bool

**Default value**  
true

When `true` duplicate frames are considered to be the ‘same’ as the previous frame and are not considered for any actions. Set to `false` to consider each frame regardless of content. This relies on the source component to only send updated frames.

**log\_file**

**Type**  
string

**Default value**  
{}

File that records what the FrameTracingComponent has done. If empty nothing is recorded.

3.6.3 Mali\_C5x\_streaming\_sink

Arm® Mali™-C55 ISP streaming output capture example component. This component tests ISP's streaming output functionality and provides an example of how to develop components consuming frames from the Mali-C55 ISP streaming output. This implementation saves all the ISP output frames to host disk in the FRM file format which is defined as part of the ISP model specification. All the implementation code is placed in this LISA+ file. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-539: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Iris and MTI instances for Mali\_C5x\_streaming\_sink

This model has the following Iris instances:

**Table 3-540: Mali\_C5x\_streaming\_sink Iris instances**

InstanceName	ComponentName
Mali_C5x_streaming_sink	Mali_C5x_streaming_sink
Mali_C5x_streaming_sink.bus_slave0	PVBusSlave
Mali_C5x_streaming_sink.bus_slave1	PVBusSlave
Mali_C5x_streaming_sink.bus_slave2	PVBusSlave

This model has the following MTI trace components:

**Table 3-541: Mali\_C5x\_streaming\_sink MTI instances**

InstanceName	ComponentName
Mali_C5x_streaming_sink.bus_slave0	PVBusSlave
Mali_C5x_streaming_sink.bus_slave1	PVBusSlave
Mali_C5x_streaming_sink.bus_slave2	PVBusSlave

## Ports for Mali\_C5x\_streaming\_sink

**Table 3-542: Ports**

Name	Protocol	Type	Description
data_s[3]	PVBus	Slave	Pixels consumer ports; receive pixels as 16-bit values (and optional debug metadata).
hsync_s	Signal	Slave	Horizontal sync port; defines input image's line begin and end for all three data ports.
vsync_s	Signal	Slave	Vertical sync port; defines the input frame's begin and end for all three data ports.

## Parameters for Mali\_C5x\_streaming\_sink

### do\_capture

#### Type

bool

#### Default value

0x1

#### Description

Saving captured frames on/off.

### fn\_prefix

#### Type

string

#### Default value

"isp\_out\_"

**Description**

Saved filenames prefix.

### 3.6.4 Mali\_C7x\_streaming\_sink

Arm® Mali™-C71/C78 ISP streaming output capture example component. This component tests ISP's streaming output functionality and provides an example of how to develop components consuming frames from the Mali-C7x ISP streaming output. This implementation saves all the ISP output frames to host disk in the FRM file format which is defined as a part of the ISP model specification. All the implementation code is placed in this LISA+ file. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-543: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Iris and MTI instances for Mali\_C7x\_streaming\_sink**

This model has the following Iris instances:

**Table 3-544: Mali\_C7x\_streaming\_sink Iris instances**

InstanceName	ComponentName
Mali_C7x_streaming_sink	Mali_C7x_streaming_sink
Mali_C7x_streaming_sink.bus_slave0	PVBusSlave
Mali_C7x_streaming_sink.bus_slave1	PVBusSlave
Mali_C7x_streaming_sink.bus_slave2	PVBusSlave

This model has the following MTI trace components:

**Table 3-545: Mali\_C7x\_streaming\_sink MTI instances**

InstanceName	ComponentName
Mali_C7x_streaming_sink.bus_slave0	PVBusSlave
Mali_C7x_streaming_sink.bus_slave1	PVBusSlave
Mali_C7x_streaming_sink.bus_slave2	PVBusSlave

**Ports for Mali\_C7x\_streaming\_sink****Table 3-546: Ports**

Name	Protocol	Type	Description
data_s[3]	PVBus	Slave	Pixels consumer ports; receive pixels as 16-bit values (and optional debug metadata).
hsync_s[3]	Signal	Slave	Horizontal sync ports; defines input image's line begin and end for the corresponding data ports.
vsync_s[3]	Signal	Slave	Vertical sync ports; define input frame's begin and end for the corresponding data port.

Parameters for Mali\_C7x\_streaming\_sink

do\_capture

Type  
bool

Default value  
0x1

Description  
Saving captured frames on/off.

fn\_prefix

Type  
string

Default value  
"isp\_out\_"

Description  
Saved filenames prefix.

3.6.5 Mali\_C55

Arm® Mali™ C55 ISP. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-547: IP revisions support

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Limitations

- Streaming input and output interfaces are supported but the protocol differs slightly from the real hardware. See Mali\_Cxx\_streaming\_camera.lisa and Mali\_C5x\_streaming\_sink.lisa for details on how to use them.
- Video monitor output and DMA monitor interface are not supported.
- Error conditions (IRQ bits 2, 3, 19, 20, 22) are not supported.
- It does not expose any targets for CADI or Iris-enabled debuggers to connect to, so debug access to its registers through CADI or Iris is not supported.

Iris and MTI instances for Mali\_C55

This model has the following Iris instances:

**Table 3-548: Mali\_C55 Iris instances**

InstanceName	ComponentName
Mali_C55	Mali_C55
Mali_C55.apb_slave	PVBusSlave
Mali_C55.bus_slave0	PVBusSlave
Mali_C55.bus_slave1	PVBusSlave
Mali_C55.bus_slave2	PVBusSlave

This model has the following MTI trace components:

**Table 3-549: Mali\_C55 MTI instances**

InstanceName	ComponentName
Mali_C55.apb_slave	PVBusSlave
Mali_C55.bus_slave0	PVBusSlave
Mali_C55.bus_slave1	PVBusSlave
Mali_C55.bus_slave2	PVBusSlave

## Ports for Mali\_C55

**Table 3-550: Ports**

Name	Protocol	Type	Description
ds_data_out_m[3]	PVBus	Master	-
ds_hsync_out_m	Signal	Master	-
ds_uv_valid_out_m	Signal	Master	-
ds_vsync_out_m	Signal	Master	-
fr_data_out_m[3]	PVBus	Master	Metadata + pixels, 16 bit, (RGB or YUV)
fr_hsync_out_m	Signal	Master	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
fr_uv_valid_out_m	Signal	Master	UV valid (set if both U and V pixels are valid)
fr_vsync_out_m	Signal	Master	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)
irq	Signal	Master	Shared interrupt
pvbus_m	PVBus	Master	Master AXI port for RAM access
pvbus_s	PVBus	Slave	Slave port for register access
reset_s	Signal	Slave	Reset signal
stream_data_in_s[3]	PVBus	Slave	Metadata + pixels, 20 bit
stream_hsync_in_s	Signal	Slave	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_vsync_in_s	Signal	Slave	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)

## Parameters for Mali\_C55

### CNR\_FITTED

#### Type

int

**Default value**

0x1

**Description**

Color Noise Reduction (CNR) and the square and square root for CNR: 0 - absent, 1 - present.

**COMPRESSION\_FITTED****Type**

int

**Default value**

0x1

**Description**

Temper compression logic: 0 - absent, 1 - present.

**DSPIPE\_FITTED****Type**

int

**Default value**

0x1

**Description**

Downscaled pipeline branch: 0 - absent, 1 - present.

**FRSCALER\_FITTED****Type**

int

**Default value**

0x1

**Description**

Rull Resolution pipeline RGB scaler: 0 - absent, 1 - present.

**IRIDIX\_GTM\_FITTED****Type**

int

**Default value**

0x0

**Description**

Iridix(TM) global tone-mapping logic: 0 - absent, 1 - present. NOTE: Must be 1 if IRIDIX\_LTM\_FITTED == 0.

**IRIDIX\_LTM\_FITTED****Type**

int

**Default value**

0x1

**Description**

Iridix(TM) local tone-mapping logic: 0 - absent, 1 - present.

**PONG\_CONFIG\_FITTED****Type**

int

**Default value**

0x1

**Description**

Pong configuration space: 0 - absent, 1 - present.

**SCALER\_COEF\_SETS****Type**

int

**Default value**

0x8

**Description**

Number of scaler coefficient sets (8 or 16).

**SINTER\_FITTED****Type**

int

**Default value**

0x1

**Description**

Sinter block: 0 - absent, 1 - present.

**SINTER\_LITE****Type**

int

**Default value**

0x0

**Description**

Sinter version: 0 - full, 1 - lite.



**TEMPER\_FITTED****Type**

int

**Default value**

0x1

**Description**

Temper, DMA, or merge: 0 - absent, 1 - present.

**WDR\_FITTED****Type**

int

**Default value**

0x1

**Description**

Wide Dynamic Range (WDR) frame stitch, offset, and gain: 0 - absent, 1 - present.

**ext\_mode****Type**

int

**Default value**

0x0

**Description**

Reserved for future use. Use it with instructions from Arm Technical Support (support-esl@arm.com).

**verbosity****Type**

int

**Default value**

0x0

**Description**

Messages verbosity level: -1 - print ERROR messages only, 0 - print ERROR and WARNING messages, 1 - print INFO messages, 2 - print DEBUG messages.

## 3.6.6 Mali\_C71

Arm® Mali™ C71 ISP. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-551: IP revisions support**

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Limitations

- Streaming input and output interfaces are supported but the protocol differs slightly from the real hardware.
- Fault Interface is not supported (except double interrupt).
- It does not expose any targets for CADI or Iris-enabled debuggers to connect to, so debug access to its registers through CADI or Iris is not supported.

### Iris and MTI instances for Mali\_C71

This model has the following Iris instances:

**Table 3-552: Mali\_C71 Iris instances**

InstanceName	ComponentName
Mali_C71	Mali_C71
Mali_C71.apb_slave	PVBusSlave
Mali_C71.bus_slave0	PVBusSlave
Mali_C71.bus_slave1	PVBusSlave
Mali_C71.bus_slave2	PVBusSlave
Mali_C71.bus_slave3	PVBusSlave

This model has the following MTI trace components:

**Table 3-553: Mali\_C71 MTI instances**

InstanceName	ComponentName
Mali_C71.apb_slave	<a href="#">PVBusSlave</a>
Mali_C71.bus_slave0	<a href="#">PVBusSlave</a>
Mali_C71.bus_slave1	<a href="#">PVBusSlave</a>
Mali_C71.bus_slave2	<a href="#">PVBusSlave</a>
Mali_C71.bus_slave3	<a href="#">PVBusSlave</a>

### Ports for Mali\_C71

**Table 3-554: Ports**

Name	Protocol	Type	Description
fault	<a href="#">Signal</a>	Master	Fault output interface
irq[4]	<a href="#">Signal</a>	Master	Shared interrupts
pvbus_m	<a href="#">PVBus</a>	Master	Master AXI port for RAM access
pvbus_s	<a href="#">PVBus</a>	Slave	Slave port for register access

Name	Protocol	Type	Description
reset_s	Signal	Slave	Reset signal.
stream_data_in_s[4]	PVBus	Slave	Metadata + pixels
stream_data_out_m[3]	PVBus	Master	Metadata + pixels
stream_hsync_in_s[4]	Signal	Slave	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_hsync_out_m[3]	Signal	Master	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_tag_in_s[4]	Value	Slave	Source stream tag Can change between lines (while hsync is Clear) if several frames are multiplexed.
stream_tag_out_m[3]	Value	Master	Source stream tag (changes while vsync is Clear)
stream_vsync_in_s[4]	Signal	Slave	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)
stream_vsync_out_m[3]	Signal	Master	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)

## Parameters for Mali\_C71

### verbosity

#### Type

int

#### Default value

0x0

#### Description

Messages verbosity level: -1 - print ERROR messages only, 0 - print ERROR and WARNING messages, 1 - print INFO messages, 2 - print DEBUG messages.

## 3.6.7 Mali\_C78

Arm® Mali™ C78 ISP. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-555: IP revisions support**

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Limitations

- Streaming input and output interfaces are supported but the protocol differs slightly from the real hardware.
- Fault Interface is not supported (except double interrupt).
- It does not expose any targets for CADI or Iris-enabled debuggers to connect to, so debug access to its registers through CADI or Iris is not supported.

## Iris and MTI instances for Mali\_C78

This model has the following Iris instances:

**Table 3-556: Mali\_C78 Iris instances**

InstanceName	ComponentName
Mali_C78	Mali_C78
Mali_C78.apb_slave	PVBusSlave
Mali_C78.bus_slave0	PVBusSlave
Mali_C78.bus_slave1	PVBusSlave
Mali_C78.bus_slave2	PVBusSlave
Mali_C78.bus_slave3	PVBusSlave

This model has the following MTI trace components:

**Table 3-557: Mali\_C78 MTI instances**

InstanceName	ComponentName
Mali_C78.apb_slave	PVBusSlave
Mali_C78.bus_slave0	PVBusSlave
Mali_C78.bus_slave1	PVBusSlave
Mali_C78.bus_slave2	PVBusSlave
Mali_C78.bus_slave3	PVBusSlave

## Ports for Mali\_C78

**Table 3-558: Ports**

Name	Protocol	Type	Description
fault	Signal	Master	Fault output interface
irq[4]	Signal	Master	Shared interrupts
pvbus_m	PVBus	Master	Master AXI port for RAM access
pvbus_s	PVBus	Slave	Slave port for register access
reset_s	Signal	Slave	Reset signal.
stream_data_in_s[4]	PVBus	Slave	Metadata + pixels
stream_data_out_m[3]	PVBus	Master	Metadata + pixels
stream_hsync_in_s[4]	Signal	Slave	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_hsync_out_m[3]	Signal	Master	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_tag_in_s[4]	Value	Slave	Source stream tag Can change between lines (while hsync is Clear) if several frames are multiplexed.
stream_tag_out_m[3]	Value	Master	Source stream tag (changes while vsync is Clear)
stream_vsync_in_s[4]	Signal	Slave	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)
stream_vsync_out_m[3]	Signal	Master	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)

## Parameters for Mali\_C78

### verbosity

#### Type

int

#### Default value

0x0

#### Description

Messages verbosity level: -1 - print ERROR messages only, 0 - print ERROR and WARNING messages, 1 - print INFO messages, 2 - print DEBUG messages.

## 3.6.8 Mali\_Cxx\_streaming\_camera

Simple streaming camera to connect to Mali ISP (C55, C71, or C78) streaming inputs. This component tests ISP's streaming input functionality and provides an example of how to develop camera-like components streaming frames to a Mali-Cxx ISP. All the implementation code is placed in this LISA+ file. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-559: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Iris and MTI instances for Mali\_Cxx\_streaming\_camera

This model has the following Iris instances:

**Table 3-560: Mali\_Cxx\_streaming\_camera Iris instances**

InstanceName	ComponentName
Mali_Cxx_streaming_camera	Mali_Cxx_streaming_camera
Mali_Cxx_streaming_camera.bus_master_data	PVBusMaster
Mali_Cxx_streaming_camera.bus_slave_config	PVBusSlave

This model has the following MTI trace components:

**Table 3-561: Mali\_Cxx\_streaming\_camera MTI instances**

InstanceName	ComponentName
Mali_Cxx_streaming_camera.bus_master_data	PVBusMaster
Mali_Cxx_streaming_camera.bus_slave_config	PVBusSlave

## Ports for Mali\_Cxx\_streaming\_camera

**Table 3-562: Ports**

Name	Protocol	Type	Description
config_s	PVBus	Slave	Access to the camera config register(s) (8 bit): 0x00 - output mode: 0 - no output, 1 - stream one frame and auto-reset to 0
data_m	PVBus	Master	Output frames port; sends pixels as 32-bit values (and debug metadata).
hsync_m	Signal	Master	Horizontal sync; set at the beginning of a line, cleared at the end.
vsync_m	Signal	Master	Vertical sync; set at the beginning of a frame, cleared at the end.

## Parameters for Mali\_Cxx\_streaming\_camera

### image\_file

#### Type

string

#### Default value

""

#### Description

A file containing one or more frames to stream in the ISP model's FRM format.

## 3.6.9 Mali\_G71

ARM Mali-G71 GPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-563: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Limitations

- The model does not support Armv8-style page tables.
- The model does not execute GPU shader programs.
- The model does not validate all register values or job descriptors.
- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.

This model outputs values for all Mali PMU hardware counters. These values are not meaningful. Their purpose is to enable you to use the model early in the development process to check that your system is correctly configured to support performance analysis tools, for instance Streamline.

The counter is a 32-bit bitfield, with the following bit assignments:

- [31:28]  
Identifies which Mali GPU instance generated this value, typically zero for a system with a single GPU.
- [27:24]  
Identifies which hardware block is the source of the counter. It can have one of the following values:  

0

Job manager.

1

Tiler.

2

L2Cache/Memory system.

3+

Shader core.
- [23:16]  
The counter number within the block.
- [15:0]  
Sawtooth counter that counts from 0 to 0xffff and then resets to 0. This value ensures that consecutive captures show different values and allows you to check that counter values are changing over time.



Note

- These counter values might change in future.
- The model always generates counter values, even if the GPU is idle.
- Streamline does not display the precise values that the model outputs because it needs to correct them for sampling frequency and other profiling effects. However, their size relative to each other is correct. For example, counters from the job manager are always smaller than those from the memory system.

Iris and MTI instances for Mali\_G71

This model has the following Iris instances:

Table 3-564: Mali\_G71 Iris instances

InstanceName	ComponentName
Mali_G71	Mali_G71
Mali_G71.busmaster	PVBusMaster
Mali_G71.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-565: Mali\_G71 MTI instances**

InstanceName	ComponentName
Mali_G71	Mali_G71
Mali_G71.busmaster	PVBusMaster
Mali_G71.busslave	PVBusSlave

## Ports for Mali\_G71

**Table 3-566: Ports**

Name	Protocol	Type	Description
gpu_reset	Signal	Slave	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	Signal	Master	The interrupt signal generated from the GPU.
irq_job	Signal	Master	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	Signal	Master	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	Signal	Master	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	PVBus	Master	The interface for the GPU to access external memory.
pvbus_s	PVBus	Slave	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G71

### revision

#### Type

string

#### Default value

"r0p0"

#### Description

Revision of the RTL that the model represents. Valid values: r0p0.

## 3.6.10 Mali\_G76

ARM Mali-G76 GPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-567: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.



## Limitations

- The model does not support Armv8-style page tables.
- The model does not execute GPU shader programs.
- The model does not validate all register values or job descriptors.
- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.

This model outputs values for all Mali PMU hardware counters. These values are not meaningful. Their purpose is to enable you to use the model early in the development process to check that your system is correctly configured to support performance analysis tools, for instance Streamline.

The counter is a 32-bit bitfield, with the following bit assignments:

### [31:28]

Identifies which Mali GPU instance generated this value, typically zero for a system with a single GPU.

### [27:24]

Identifies which hardware block is the source of the counter. It can have one of the following values:

**0**

Job manager.

**1**

Tiler.

**2**

L2Cache/Memory system.

**3+**

Shader core.

### [23:16]

The counter number within the block.

### [15:0]

Sawtooth counter that counts from 0 to 0xffff and then resets to 0. This value ensures that consecutive captures show different values and allows you to check that counter values are changing over time.



- These counter values might change in future versions.
- The model always generates counter values, even if the GPU is idle.
- Streamline does not display the precise values that the model outputs because it needs to correct them for sampling frequency and other profiling effects. However, their size relative to each other is correct. For example, counters from the job manager are always smaller than those from the memory system.

## Iris and MTI instances for Mali\_G76

This model has the following Iris instances:

**Table 3-568: Mali\_G76 Iris instances**

InstanceName	ComponentName
Mali_G76	Mali_G76
Mali_G76.busmaster	PVBusMaster
Mali_G76.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-569: Mali\_G76 MTI instances**

InstanceName	ComponentName
Mali_G76	Mali_G76
Mali_G76.busmaster	PVBusMaster
Mali_G76.busslave	PVBusSlave

## Ports for Mali\_G76

**Table 3-570: Ports**

Name	Protocol	Type	Description
gpu_reset	Signal	Slave	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	Signal	Master	The interrupt signal generated from the GPU.
irq_job	Signal	Master	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	Signal	Master	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	Signal	Master	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	PVBus	Master	The interface for the GPU to access external memory.
pvbus_s	PVBus	Slave	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G76

### revision

#### Type

string

#### Default value

"rOp0"

#### Description

Revision of the RTL that the model represents. Valid values: rOp0.

### 3.6.11 Mali\_G78AE

Arm® Mali™-G78AE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-571: IP revisions support**

Revision	Quality level
r0p0 r0p1 r0p2	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About Mali\_G78AE

The Mali\_G78AE component models the Arm Mali G78AE, which implements the Valhall architecture, and is the first Mali GPU designed specifically to target automotive use cases.

The model implements the Partition Manager, enabling up to 4 independent partitions running workloads at the same time, while being accessed by up to 16 virtual machines.

The Mali\_G78AE model is functional, capable of executing GPU shader programs and producing graphical or compute outputs. It requires a clock input running at around 500MHz to keep the CPU and GPU performance aligned, and to avoid software timeouts during slow-running GPU operations.

With this model, and the implemented functionality of the Partition Manager, you can:

- Simulate the entire graphics software stack, including the user space and kernel space driver components of the Arm Mali DDK, and an application that uses a graphics API.
- Verify the integration of the Mali G78AE GPU into the rest of the platform. A complex use case example of this is a system running multiple kernels in Virtual Machines under a Hypervisor, all submitting workloads to the GPU at the same time.

To configure the reference Mali driver for use with the model, we recommend you make some adjustments to the timing parameters of the Mali driver. This is because of timing differences between the real hardware and the Fast Model. Which parameters work best depend on your system, but if you are using the reference Arm implementation of the arbiter from the DDK, one possibility is:

```
insmod mali_kbase.ko gpu_req_timeout=1000
insmod mali_arbiter.ko request_timeout=200 yield_timeout=30
```

#### Test applications

The tests have been carried out using the r40p0 release of the Mali DDK. The following applications have been tested and confirmed to work:

- A selection of the Mali DDK integration tests in `product/build-<wsi>/install/bin/â€`
  - mali\_gles\_integration\_suite
  - mali\_cl\_simple\_example

- A selection of lightweight Vulkan examples hosted on [GitHub](#):
  - gears
  - computeparticles, at small particle count
  - texture3d
- A selection of ComputeLibrary examples hosted on [GitHub](#):
  - graph\_lenet
  - graph\_mobilenet\_v2

All of these applications have been successfully tested in a non-virtualised system, on Debian Buster running Linux 4.19.

A subset of these examples have been tested in the following virtualised systems:

- Xen Hypervisor 4.14.1-pre
- Privileged Debian Buster running Linux kernel 4.19, controlling the configuration of partitions, and running the Mali DDK reference arbiter implementation
- Two unprivileged virtual machines, both running Debian Buster, Linux 4.19, both running various applications listed in this section at the same time

## Limitations

- The model does not support debug access to registers through CADI or Iris-enabled debuggers.
- It is not supported on Windows hosts.
- It is a functional model and does not simulate performance differences for partitions of different sizes.
- It does not implement the protection \*CHK signals.
- It does not implement Parity/DCLS/CRC faults and fault fingerprints.
- It does not implement different error response modes configurable through the SYSTEM page.
- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.

## Iris and MTI instances for Mali\_G78AE

This model has the following Iris instances:

**Table 3-572: Mali\_G78AE Iris instances**

InstanceName	ComponentName
Mali_G78AE	Mali_G78AE
Mali_G78AE.AccessControl	PVBusMapper

This model has the following MTI trace components:

**Table 3-573: Mali\_G78AE MTI instances**

InstanceName	ComponentName
Mali_G78AE	Mali_G78AE
Mali_G78AE.AccessControl	PVBusMapper

### Ports for Mali\_G78AE

**Table 3-574: Ports**

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
gpu_reset	Signal	Slave	Reset signals
gpu_reset_recovery	Signal	Slave	-
irq_deferred_error	Signal	Master	-
irq_groups[4]	Signal	Master	-
irq_partitions[4]	Signal	Master	Partition Manager irqs
irq_uncorrected_error	Signal	Master	-
irq_windows[16]	Signal	Master	-
pvbus_m	PVBus	Master	Output to board from GPU
pvbus_s[3]	PVBus	Slave	Slave bus ports (AXI-A through C).
sys_assign_enable	Value	Slave	System configuration access control for different ports

### Parameters for Mali\_G78AE

#### labeller\_encoding\_spec

##### Type

string

##### Default value

"MasterID[31:0]=StreamID[31:0]"

##### Description

Specification of how the StreamID is encoded into transaction attributes.

#### revision

##### Type

string

##### Default value

"r0p0"

##### Description

Hardware revision. Changing this parameter aligns the behaviour of the model with the hardware of specified revision.

### 3.6.12 Mali\_G710

Arm® Mali™ G710 GPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-575: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About Mali\_G710

The Mali\_G710 and Mali\_G715 components model the Arm Mali G710 and G715 GPUs respectively, which implement the Valhall architecture. They are among the first Mali GPUs to implement a Command Stream Frontend (CSF), which is a combination of hardware and firmware that helps to offload work from the Mali driver that is running on the CPU.

With the Mali G710 and G715 Fast Models, you can simulate:

- User space and kernel space driver components of the Arm Mali Driver Development kit (Mali DDK), when built to target Mali G710 or G715 under Android and Linux graphics stacks.
- Mali G710 or G715 firmware binaries.

These are functional models that do not attempt to execute GPU shader programs. They are sufficient to simulate a Mali driver, but rather than deliver fully rendered images to memory, they write a randomly-generated color gradient pattern to the framebuffer.

The changing frames can be used to prove that the graphics stack is correctly configured to allow the Mali GPU to write to a user-visible framebuffer.

#### Configuring the driver for use with the model

To account for timing differences between real hardware and the Fast Model, it is necessary to modify the following runtime settings in the Mali driver:

##### **csf\_firmware\_boot\_timeout\_ms**

Overrides the minimum timeout value for loading firmware into the model.

##### **Type**

Kernel module parameter

##### **Recommended value**

10000

##### **reset\_timeout**

Overrides the minimum timeout when waiting for operations on the GPU to finish.

##### **Type**

Sysfs parameter

**Recommended value**

1000000

**fw\_timeout**

Overrides the minimum timeout when waiting for operations on the GPU to finish.

**Type**

Sysfs parameter

**Recommended value**

1000000

The following commands are a snippet from an Android RC file that is an example of how a filesystem can be configured to correctly load the Mali driver when running on a Fast Models platform:

```
insmod /vendor/lib/modules/mali_kbase.ko csf_firmware_boot_timeout_ms=100000
wait /dev/mali0
chmod 0666 /dev/mali0
wait /sys/class/misc/mali0/device/reset_timeout
write /sys/class/misc/mali0/device/reset_timeout 1000000
write /sys/class/misc/mali0/device/fw_timeout 1000000
```

**Limitations**

- The model does not support debug access to registers through CADI or Iris-enabled debuggers.
- The timing data observed from the model does not correspond in any way to real hardware.
- The model does not expose any MTI targets.
- The model is not supported on Windows hosts.
- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.

**Iris and MTI instances for Mali\_G710**

This model has the following Iris instances:

**Table 3-576: Mali\_G710 Iris instances**

InstanceName	ComponentName
Mali_G710	Mali_G710
Mali_G710.busmaster	PVBusMaster
Mali_G710.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-577: Mali\_G710 MTI instances**

InstanceName	ComponentName
Mali_G710	Mali_G710
Mali_G710.busmaster	PVBusMaster

InstanceName	ComponentName
Mali_G710.busslave	PVBusSlave

## Ports for Mali\_G710

**Table 3-578: Ports**

Name	Protocol	Type	Description
gpu_reset	Signal	Slave	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	Signal	Master	The interrupt signal generated from the GPU.
irq_job	Signal	Master	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	Signal	Master	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	Signal	Master	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	PVBus	Master	The interface for the GPU to access external memory.
pvbus_s	PVBus	Slave	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G710

### altcmdline

#### Type

string

#### Default value

""

#### Description

Alternate command line for the GPU model. If used, mode is ignored.

### altmodel

#### Type

string

#### Default value

""

#### Description

Path to an alternative GPU model library.

### mode

#### Type

string

#### Default value

"fast"



**Description**

GPU Mode. Inputs supported: [fast].

**revision**

**Type**

string

**Default value**

"r0p0"

**Description**

Hardware revision. Changing this parameter aligns the behaviour of the model with the hardware of specified revision.

3.6.13 Mali\_G715

Arm® Mali™ G715 GPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-579: IP revisions support

Revision	Quality level
r0p0 r0p1 r1p0 r1p1 r1p2 r1p3	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About Mali\_G715

The Mali\_G710 and Mali\_G715 components model the Arm Mali G710 and G715 GPUs respectively, which implement the Valhall architecture. They are among the first Mali GPUs to implement a Command Stream Frontend (CSF), which is a combination of hardware and firmware that helps to offload work from the Mali driver that is running on the CPU.

With the Mali G710 and G715 Fast Models, you can simulate:

- User space and kernel space driver components of the Arm Mali Driver Development kit (Mali DDK), when built to target Mali G710 or G715 under Android and Linux graphics stacks.
- Mali G710 or G715 firmware binaries.

These are functional models that do not attempt to execute GPU shader programs. They are sufficient to simulate a Mali driver, but rather than deliver fully rendered images to memory, they write a randomly-generated color gradient pattern to the framebuffer.

The changing frames can be used to prove that the graphics stack is correctly configured to allow the Mali GPU to write to a user-visible framebuffer.

## Configuring the driver for use with the model

To account for timing differences between real hardware and the Fast Model, it is necessary to modify the following runtime settings in the Mali driver:

### **csf\_firmware\_boot\_timeout\_ms**

Overrides the minimum timeout value for loading firmware into the model.

#### **Type**

Kernel module parameter

#### **Recommended value**

10000

### **reset\_timeout**

Overrides the minimum timeout when waiting for operations on the GPU to finish.

#### **Type**

Sysfs parameter

#### **Recommended value**

1000000

### **fw\_timeout**

Overrides the minimum timeout when waiting for operations on the GPU to finish.

#### **Type**

Sysfs parameter

#### **Recommended value**

1000000

The following commands are a snippet from an Android RC file that is an example of how a filesystem can be configured to correctly load the Mali driver when running on a Fast Models platform:

```
insmod /vendor/lib/modules/mali_kbase.ko csf_firmware_boot_timeout_ms=100000
wait /dev/mali0
chmod 0666 /dev/mali0
wait /sys/class/misc/mali0/device/reset_timeout
write /sys/class/misc/mali0/device/reset_timeout 1000000
write /sys/class/misc/mali0/device/fw_timeout 1000000
```

## Limitations

- The model does not support debug access to registers through CADI or Iris-enabled debuggers.
- The timing data observed from the model does not correspond in any way to real hardware.
- The model does not expose any MTI targets.
- The model is not supported on Windows hosts.
- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.

## Iris and MTI instances for Mali\_G715

This model has the following Iris instances:

**Table 3-580: Mali\_G715 Iris instances**

InstanceName	ComponentName
Mali_G715	Mali_G715
Mali_G715.busmaster	PVBusMaster
Mali_G715.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-581: Mali\_G715 MTI instances**

InstanceName	ComponentName
Mali_G715	Mali_G715
Mali_G715.busmaster	PVBusMaster
Mali_G715.busslave	PVBusSlave

## Ports for Mali\_G715

**Table 3-582: Ports**

Name	Protocol	Type	Description
gpu_reset	Signal	Slave	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	Signal	Master	The interrupt signal generated from the GPU.
irq_job	Signal	Master	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	Signal	Master	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	Signal	Master	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	PVBus	Master	The interface for the GPU to access external memory.
pvbus_s	PVBus	Slave	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G715

### altcmdline

#### Type

string

#### Default value

""

#### Description

Alternate command line for the GPU model. If used, mode is ignored.

**altmodel**

**Type**  
string

**Default value**  
""

**Description**  
Path to an alternative GPU model library.

**mode**

**Type**  
string

**Default value**  
"fast"

**Description**  
GPU Mode. Inputs supported: [fast].

**revision**

**Type**  
string

**Default value**  
"r0p0"

**Description**  
Hardware revision. Changing this parameter aligns the behaviour of the model with the hardware of specified revision.

3.6.14 Mali\_G720

Arm® Mali™ G720 GPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-583: IP revisions support

Revision	Quality level
r0p0 r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Changes in 11.29.19

Ports added:

- cntvalueb

## About Mali\_G720

This model is the first of a new generation of fully-functional GPU models that generate correct output without the assistance of Generic Graphics Accelerator (GGA).

It supports x86\_64 and AArch64 hosts running a supported version of Linux, as listed in Requirements for Fast Models in the [Fast Models User Guide](#). It does not support Windows hosts.

`$(PVLIB_HOME)/LISA/Mali_G720.lisa` requires a clock input running at around 500MHz to keep CPU and GPU performance aligned. This avoids software timeouts during slow-running GPU operations.

From a Fast Models perspective, the differences between Arm Mali-G720, Arm Immortalis-G720, and Mali-G620 are minor. The Fast Model does not model the caches and the core count is transparent to the user. Ray tracing is not currently modeled.

## Limitations

- The model does not support debug access to registers through CADI or Iris-enabled debuggers.
- The timing data observed from the model does not correspond in any way to real hardware.
- The model is not supported on Windows hosts.

## Iris and MTI instances for Mali\_G720

This model has the following Iris instances:

**Table 3-584: Mali\_G720 Iris instances**

InstanceName	ComponentName
Mali_G720	Mali_G720
Mali_G720.busmaster	PVBusMaster
Mali_G720.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-585: Mali\_G720 MTI instances**

InstanceName	ComponentName
Mali_G720	Mali_G720
Mali_G720.busmaster	PVBusMaster
Mali_G720.busslave	PVBusSlave

## Ports for Mali\_G720

**Table 3-586: Ports**

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-

Name	Protocol	Type	Description
cntvalueb	CounterInterface	Slave	Interface to the SoC-level Generic Counter module that is used to source the global system timestamp value. When the GPU component is integrated into an A-Profile compliant system, this port must be connected for the GPU and CPU timestamps to be synchronized. Without the connection, the GPU timestamp remains constant. In this case, it is recommended to configure the GPU to use its own internal clock as the reference.
gpu_reset	Signal	Slave	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	Signal	Master	The interrupt signal generated from the GPU.
irq_job	Signal	Master	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	Signal	Master	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	Signal	Master	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	PVBus	Master	The interface for the GPU to access external memory.
pvbus_s	PVBus	Slave	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G720

### altcmdline

#### Type

string

#### Default value

""

#### Description

Alternate command line for the GPU model. If used, mode is ignored.

### altmodel

#### Type

string

#### Default value

""

#### Description

Path to an alternative GPU model library.

### mode

#### Type

string

#### Default value

"fast"

#### Description

GPU Mode. Inputs supported: [fast, turbo\_fallback, turbo].

**revision**

**Type**

string

**Default value**

"r0p0"

**Description**

Hardware revision. Changing this parameter aligns the behaviour of the model with the hardware of specified revision.

**turbo\_threads**

**Type**

int

**Default value**

0x0

**Description**

Number of threads used.

3.6.15 Mali\_G725

Arm® Mali™ G725 GPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-587: IP revisions support

Revision	Quality level
r0p0 r0p1	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Changes in 11.29.19

Ports added:

- cntvalueb

Parameters removed:

- turbo\_threads

About Mali\_G725

This component is a model of the Arm Mali-G725 GPU, fully capable of executing shaders and producing graphical or compute outputs.

It supports x86\_64 and AArch64 hosts running a supported version of Linux, as listed in Requirements for Fast Models in the [Fast Models User Guide](#). It does not support Windows hosts.

`$(PVLIB_HOME)/LISA/Mali_G725.lisa` requires a clock input running at around 500MHz to keep CPU and GPU performance aligned. This avoids software timeouts during slow-running GPU operations.

From a Fast Models perspective, the differences between Mali-G725, Arm Immortalis-G925, and Mali-G625 are minor. The Fast Model does not model the caches and the core count is transparent to the user. Ray tracing is not currently modeled.

## Limitations

- The model does not support debug access to registers through CADI or Iris-enabled debuggers.
- The timing data observed from the model does not correspond in any way to real hardware.
- The model is not supported on Windows hosts.

## Iris and MTI instances for Mali\_G725

This model has the following Iris instances:

**Table 3-588: Mali\_G725 Iris instances**

InstanceName	ComponentName
Mali_G725	Mali_G725
Mali_G725.busmaster	PVBusMaster
Mali_G725.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-589: Mali\_G725 MTI instances**

InstanceName	ComponentName
Mali_G725	Mali_G725
Mali_G725.busmaster	PVBusMaster
Mali_G725.busslave	PVBusSlave

## Ports for Mali\_G725

**Table 3-590: Ports**

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
cntvalueb	CounterInterface	Slave	Interface to the SoC-level Generic Counter module that is used to source the global system timestamp value. When the GPU component is integrated into an A-Profile compliant system, this port must be connected for the GPU and CPU timestamps to be synchronized. Without the connection, the GPU timestamp remains constant. In this case, it is recommended to configure the GPU to use its own internal clock as the reference.
gpu_reset	Signal	Slave	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	Signal	Master	The interrupt signal generated from the GPU.
irq_job	Signal	Master	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	Signal	Master	The interrupt signal generated from the MMU on the GPU.



Name	Protocol	Type	Description
prot_mode_m0	Signal	Master	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbush_m port.
pvbus_m	PVBus	Master	The interface for the GPU to access external memory.
pvbus_s	PVBus	Slave	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G725

### altcmdline

#### Type

string

#### Default value

""

#### Description

Alternate command line for the GPU model. If used, mode is ignored.

### altmodel

#### Type

string

#### Default value

""

#### Description

Path to an alternative GPU model library.

### mode

#### Type

string

#### Default value

"fast"

#### Description

GPU Mode. Inputs supported: [fast].

### revision

#### Type

string

#### Default value

"r0p0"

## Description

Hardware revision. Changing this parameter aligns the behaviour of the model with the hardware of specified revision.

## 3.6.16 Mali\_T624

ARM Mali-T624 GPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-591: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Limitations

- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.

## Iris and MTI instances for Mali\_T624

This model has the following Iris instances:

**Table 3-592: Mali\_T624 Iris instances**

InstanceName	ComponentName
Mali_T624	Mali_T624
Mali_T624.busmaster	PVBusMaster
Mali_T624.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-593: Mali\_T624 MTI instances**

InstanceName	ComponentName
Mali_T624	Mali_T624
Mali_T624.busmaster	PVBusMaster
Mali_T624.busslave	PVBusSlave

## Ports for Mali\_T624

**Table 3-594: Ports**

Name	Protocol	Type	Description
gpu_reset	Signal	Slave	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	Signal	Master	The interrupt signal generated from the GPU.
irq_job	Signal	Master	The interrupt signal generated from the Job Manager on the GPU.

Name	Protocol	Type	Description
irq_mmu	Signal	Master	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	Signal	Master	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbuss_m port.
pvbuss_m	PVBus	Master	The interface for the GPU to access external memory.
pvbuss_s	PVBus	Slave	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_T624

### revision

#### Type

string

#### Default value

"r0p0"

#### Description

Revision of the RTL that the model represents. Valid values: r0p0.

## 3.6.17 V61

Arm® Mali™-V61 Video Processor. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-595: IP revisions support**

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About V61

The model requires an external OpenMAX (OMX) IL implementation for codec functionality. By default, V61 looks for `ffomaxil.dll` on Windows or `libffomaxil.so` on Linux in the model binary's directory or in the Fast Models installation. The default library path can be overridden using the `omx-library-path` parameter.

FFomaxIL is an OMX IL implementation provided by Arm in the TPIP package for convenience. Refer to the TPIP package for more details on FFomaxIL. It is available on [Product Download Hub](#)

When querying the OMX core, V61 searches for the following roles in the list of OpenMAX components:

#### H.264 decode

"video\_decoder.avc"

**JPEG decode**`"video_decoder.mjpeg"`**MPEG2 decode**`"video_decoder.mpeg2"`**MPEG4 decode**`"video_decoder.mpeg4"`**VC1 decode**`"video_decoder.vc1"`**VP8 decode**`"video_decoder.vp8"`**VP8 encode**`"video_encoder.vp8"`**Note**

To build example platforms containing V61, you must either install the TPIP package, or remove the dependency on FFmpeg and libvpx from the platform's `sgproj` file, by removing the line containing `v5xx.sgrepo`.

**Limitations**

- No support for HEVC, VP9 and RealVideo decoders.
- No support for 10-bit video output.
- No support for RGB or AFBC input for encoding.
- No profiling support.
- No QoS support.
- Power/Test modes are modeled only as register state changes.

**Iris and MTI instances for V61**

This model has the following Iris instances:

**Table 3-596: V61 Iris instances**

InstanceName	ComponentName
V61	V61
V61.BusModifier.LSID0	PVBusMapper
V61.BusModifier.LSID1	PVBusMapper
V61.BusModifier.LSID2	PVBusMapper
V61.BusModifier.LSID3	PVBusMapper
V61.apb_slave[0]	PVBusSlave

This model has the following MTI trace components:

**Table 3-597: V61 MTI instances**

InstanceName	ComponentName
V61	V61
V61.BusModifier.LSID0	PVBusMapper
V61.BusModifier.LSID1	PVBusMapper
V61.BusModifier.LSID2	PVBusMapper
V61.BusModifier.LSID3	PVBusMapper
V61.apb_slave[0]	PVBusSlave

## Ports for V61

**Table 3-598: Ports**

Name	Protocol	Type	Description
apb_s	PVBus	Slave	APB Slave port for register access.
axi_m	PVBus	Master	AXI master bus for memory accesses.
clk	ClockSignal	Slave	Master clock, typically 300MHz.
irq	Signal	Master	IRQ signal to host CPU.
reset	Signal	Slave	Reset signal.

## Parameters for V61

### AXI-data-width

#### Type

int

#### Default value

0x4

#### Description

AXI data width, logarithmic byte notation (3-&gt;64bit, 4-&gt;128 bit).

### enable-frame-rate-limiting

#### Type

bool

#### Default value

0x0

#### Description

enable output rate control via the CLK port.

### fuse-disable-AFBC

#### Type

bool

**Default value**

0x0

**Description**

disable AFBC support by fuse.

**`fuse-disable-HEVC`****Type**

bool

**Default value**

0x0

**Description**

disable HEVC support by fuse.

**`fuse-disable-Real`****Type**

bool

**Default value**

0x0

**Description**

disable RealVideo support by fuse.

**`fuse-disable-VPX`****Type**

bool

**Default value**

0x0

**Description**

disable VP8 support by fuse.

**`ncores`****Type**

int

**Default value**

0x1

**Description**

Number of cores in the component.

**`omx-library-path`****Type**

string

**Default value**

""

**Description**

path to a user-provided OMX library; leave blank to use FFomaxIL.

**supports-10bit****Type**

bool

**Default value**

0x1

**Description**

component supports 10-bit content decoding.

**supports-64byte-ref-bursts****Type**

bool

**Default value**

0x1

**Description**

component supports 64-byte bursts for reference pixel data.

**supports-encoding****Type**

bool

**Default value**

0x1

**Description**

component supports encoding.

## 3.7 Peripheral components

This section describes the Peripheral components.

### 3.7.1 Ashbrook\_SoC\_SCC

Ashbrook SoC Simple Configuration Controller IP Block. This model is written in C++.

#### Iris and MTI instances for Ashbrook\_SoC\_SCC

This model has the following Iris instances:

**Table 3-599: Ashbrook\_SoC\_SCC Iris instances**

InstanceName	ComponentName
Ashbrook_SoC_SCC	Ashbrook_SoC_SCC
Ashbrook_SoC_SCC.pvbuslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-600: Ashbrook\_SoC\_SCC MTI instances**

InstanceName	ComponentName
Ashbrook_SoC_SCC.pvbuslave	PVBusSlave

### Ports for Ashbrook\_SoC\_SCC

**Table 3-601: Ports**

Name	Protocol	Type	Description
cluster_temperature[16]	ValueState	Slave	-
pvbus_s	PVBus	Slave	-

### Parameters for Ashbrook\_SoC\_SCC

#### diagnostics

##### Type

int

##### Default value

0x0

##### Description

Diagnostics.

#### gpr0

##### Type

int

##### Default value

0x0

##### Description

General Purpose Register 0.

#### gpr1

##### Type

int

##### Default value

0x0



**Description**

General Purpose Register 1.

**gpr10****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 10.

**gpr11****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 11.

**gpr12****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 12.

**gpr13****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 13.

**gpr14****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 14.

**gpr15****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 15.

**gpr2****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 2.

**gpr3****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 3.

**gpr4****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 4.

**gpr5****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 5.

**gpr6****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 6.

**gpr7****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 7.

**gpr8****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 8.

**gpr9****Type**

int

**Default value**

0x0

**Description**

General Purpose Register 9.

**num\_clusters****Type**

int

**Default value**

0xc

**Description**  
Number of AP clusters in the platform.

**version**

**Type**  
int

**Default value**  
0x0

**Description**  
Version number of the platform.

3.7.2 AudioOut\_File

File based Audio Output for PL041\_AACI. This model is written in LISA+.

**About AudioOut\_File**  
This component implements an audio output that is suitable for use with the PL041\_AACI component. It writes raw 16-bit 48KHz stereo audio data to a user-specified file.  
  
We expect this component to have little effect on the performance of PV systems. AudioOut\_File drains audio data at the rate that would be expected by software running in the simulation.

**Iris and MTI instances for AudioOut\_File**  
This model has the following Iris instances:

Table 3-602: AudioOut\_File Iris instances

InstanceName	ComponentName
AudioOut_File	AudioOut_File

Ports for AudioOut\_File

Table 3-603: Ports

Name	Protocol	Type	Description
audio	AudioControl	Slave	Audio input for a connection to a component such as the PL041_AACI.

Parameters for AudioOut\_File

**fname**  
**Type**  
string  
**Default value**  
""

**Description**

Filename.

### 3.7.3 AudioOut\_SDL

SDL based Audio Output for PL041\_AACI. This model is written in LISA+.

**About AudioOut\_SDL**

AudioOut\_SDL outputs audio using the host features of the Simple DirectMedia Layer (SDL) library.

This component results in SDL audio callbacks and might have a small impact on PV systems containing the component. It attempts to drain audio data at whatever rate is required to maintain smooth sound playback on the host PC. This might not match the data rate expected by applications running on the simulation.

**Iris and MTI instances for AudioOut\_SDL**

This model has the following Iris instances:

**Table 3-604: AudioOut\_SDL Iris instances**

InstanceName	ComponentName
AudioOut_SDL	AudioOut_SDL

**Ports for AudioOut\_SDL****Table 3-605: Ports**

Name	Protocol	Type	Description
audio	AudioControl	Slave	Audio input for a connection to a component such as the PL041_AACI.

### 3.7.4 Base\_PowerController

Base Platforms Power Controller. This model is written in LISA+.

**Iris and MTI instances for Base\_PowerController**

This model has the following Iris instances:

**Table 3-606: Base\_PowerController Iris instances**

InstanceName	ComponentName
Base_PowerController	Base_PowerController
Base_PowerController.busslave	PVBusSlave
Base_PowerController.timer_ppu_transition	ClockTimerThread
Base_PowerController.timer_ppu_transition.timer	ClockTimerThread64
Base_PowerController.timer_ppu_transition.timer.thread	SchedulerThread
Base_PowerController.timer_ppu_transition.timer.thread_event	SchedulerThreadEvent

InstanceName	ComponentName
Base_PowerController.timer_reset	ClockTimerThread
Base_PowerController.timer_reset.timer	ClockTimerThread64
Base_PowerController.timer_reset.timer.thread	SchedulerThread
Base_PowerController.timer_reset.timer.thread_event	SchedulerThreadEvent
Base_PowerController.timer_sys_reset_request	ClockTimerThread
Base_PowerController.timer_sys_reset_request.timer	ClockTimerThread64
Base_PowerController.timer_sys_reset_request.timer.thread	SchedulerThread
Base_PowerController.timer_sys_reset_request.timer.thread_event	SchedulerThreadEvent
Base_PowerController.utility_bus0	PVBusMaster
Base_PowerController.utility_bus1	PVBusMaster
Base_PowerController.utility_bus2	PVBusMaster
Base_PowerController.utility_bus3	PVBusMaster

This model has the following MTI trace components:

**Table 3-607: Base\_PowerController MTI instances**

InstanceName	ComponentName
Base_PowerController	Base_PowerController
Base_PowerController.busslave	PVBusSlave
Base_PowerController.utility_bus0	PVBusMaster
Base_PowerController.utility_bus1	PVBusMaster
Base_PowerController.utility_bus2	PVBusMaster
Base_PowerController.utility_bus3	PVBusMaster

### Ports for Base\_PowerController

**Table 3-608: Ports**

Name	Protocol	Type	Description
cpuporeset[42]	Signal	Master	-
dbgnopwrdown[42]	Signal	Slave	-
l2reset[4]	Signal	Master	-
pchannel_m[42]	PChannel	Master	-
pvbus_s	PVBus	Slave	-
standbywfi[42]	Signal	Slave	-
standbywfi12[4]	Signal	Slave	-
system_reset	Signal	Master	-
system_reset_req	Signal	Slave	-
utility_bus_m[4]	PVBus	Master	-
wakerequest[42]	Signal	Slave	-

## Parameters for Base\_PowerController

### **Affinity-shifted**

**Type**

bool

**Default value**

0x0

**Description**

Whether core number is reflected in Affinity1 instead of Affinity0.

### **CPU-affinities**

**Type**

string

**Default value**

"0.0.0.0"

**Description**

Definition of which cores are attached to the control pins, as a comma separated list of affinity dotted quads.

### **CPU-available-mask**

**Type**

int

**Default value**

0xffffffffffffffffffff

**Description**

One bit per entry in CPU-affinities list, set zero if a CPU is wired up but actually not available.

### **enable\_lock\_step**

**Type**

bool

**Default value**

0x0

**Description**

If lock step is enabled, the number of available cores get reduced to half.

### **startup**

**Type**

string

**Default value**

"0.0.0.\*"

**Description**

Comma-separated list of cores (wildcards allowed) to be powered up at startup or system reset.

**use\_in\_cluster\_ppu****Type**

bool

**Default value**

0x0

**Description**

Set this to true if base power controller is connected to V9 core where in-cluster PPU is used, false, otherwise.

**use\_pchannel\_for\_threads****Type**

bool

**Default value**

0x0

**Description**

Set this to true if the pchannel is connected to cpus with thread support.

### 3.7.5 CMSDK\_Timer

ARM Timer Module. This model is written in C++.

**Iris and MTI instances for CMSDK\_Timer**

This model has the following Iris instances:

**Table 3-609: CMSDK\_Timer Iris instances**

InstanceName	ComponentName
CMSDK_Timer	CMSDK_Timer
CMSDK_Timer.busslave	PVBusSlave
CMSDK_Timer.clk_div	ClockDivider
CMSDK_Timer.counter	CounterModule

This model has the following MTI trace components:

**Table 3-610: CMSDK\_Timer MTI instances**

InstanceName	ComponentName
CMSDK_Timer.busslave	<a href="#">PVBusSlave</a>
CMSDK_Timer.clk_div	<a href="#">ClockDivider</a>



## Ports for CMSDK\_Timer

Table 3-611: Ports

Name	Protocol	Type	Description
clock	<a href="#">ClockSignal</a>	Slave	-
irq_out	<a href="#">Signal</a>	Master	-
pvbuss	<a href="#">PVBuss</a>	Slave	-

## Parameters for CMSDK\_Timer

**clk\_div.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clk\_div.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**counter.diagnostics****Type**

int

**Default value**

0x2

**Description**

Diagnostics.

**diagnostics****Type**

int

**Default value**

0x2

**Description**

Diagnostics.

### 3.7.6 Clock\_Multiplexer

Clock Multiplexer. This model is written in C++.

#### Iris and MTI instances for Clock\_Multiplexer

This model has the following Iris instances:

**Table 3-612: Clock\_Multiplexer Iris instances**

InstanceName	ComponentName
Clock_Multiplexer	Clock_Multiplexer

#### Ports for Clock\_Multiplexer

**Table 3-613: Ports**

Name	Protocol	Type	Description
cur_clkssel	Value	Master	-
input[16]	ClockSignal	Slave	-
output	ClockSignal	Master	-
selector	Value	Slave	-
stopclk_nominal	Value	Slave	-

#### Parameters for Clock\_Multiplexer

##### diagnostics

##### Type

int

##### Default value

0x2

##### Description

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

### 3.7.7 Cluster\_Temperature\_Sensor

Component to calculate the temperature value of all cores in a cluster. This model is written in C++.

#### Iris and MTI instances for Cluster\_Temperature\_Sensor

This model has the following Iris instances:

**Table 3-614: Cluster\_Temperature\_Sensor Iris instances**

InstanceName	ComponentName
Cluster_Temperature_Sensor	Cluster_Temperature_Sensor

## Ports for Cluster\_Temperature\_Sensor

Table 3-615: Ports

Name	Protocol	Type	Description
cluster_powerdown_in	Signal	Slave	-
core_powerdown_in[16]	Signal	Slave	-
core_state_in[16]	ValueState	Slave	-
core_ticks_in[16]	InstructionCount	Slave	-
freq_in	ValueState	Slave	-
temperature_out	ValueState	Master	-

## Parameters for Cluster\_Temperature\_Sensor

**MAX\_FREQ**

## Type

int

## Default value

0x77359400

## Description

Maximum frequency (in Hz) at which each core can run.

**NUM\_CORES**

## Type

int

## Default value

0x4

## Description

Number of cores per cluster.

**diagnostics**

## Type

int

## Default value

0x0

## Description

Diagnostics.

**tempCoeff\_A**

## Type

string

**Default value**  
"0.5"

**Description**  
Temperature Coefficient.

**tempCoeff\_B**

**Type**  
string

**Default value**  
"0.5"

**Description**  
Temperature Coefficient.

**tempCoeff\_K**

**Type**  
int

**Default value**  
0x32

**Description**  
Temperature Coefficient.

**tempCoeff\_TAMB**

**Type**  
int

**Default value**  
0x14

**Description**  
Temperature Coefficient.

3.7.8 CombinedMessagingUnit

CMU - Combined MHU monolithic block. This model is written in LISA+.

Iris and MTI instances for CombinedMessagingUnit

This model has the following Iris instances:

Table 3-616: CombinedMessagingUnit Iris instances

InstanceName	ComponentName
CombinedMessagingUnit	CombinedMessagingUnit
CombinedMessagingUnit.host_to_local	MessageHandlingUnit

InstanceName	ComponentName
CombinedMessagingUnit.host_to_local.a_to_b_v2	MessageHandlingUnitV2
CombinedMessagingUnit.host_to_local.a_to_b_v3	MessageHandlingUnitV3
CombinedMessagingUnit.host_to_local.version_mapper_a_to_b_rec	PVBusMapper
CombinedMessagingUnit.host_to_local.version_mapper_a_to_b_snd	PVBusMapper
CombinedMessagingUnit.host_to_local_rcv_log	PVBusLogger
CombinedMessagingUnit.host_to_local_rcv_log.mapper	PVBusMapper
CombinedMessagingUnit.host_to_local_snd_log	PVBusLogger
CombinedMessagingUnit.host_to_local_snd_log.mapper	PVBusMapper
CombinedMessagingUnit.irq_rcv_combined_host_log	SignalLogger
CombinedMessagingUnit.irq_rcv_combined_local_log	SignalLogger
CombinedMessagingUnit.irq_snd_combined_host_log	SignalLogger
CombinedMessagingUnit.irq_snd_combined_local_log	SignalLogger
CombinedMessagingUnit.local_to_host	MessageHandlingUnit
CombinedMessagingUnit.local_to_host.a_to_b_v2	MessageHandlingUnitV2
CombinedMessagingUnit.local_to_host.a_to_b_v3	MessageHandlingUnitV3
CombinedMessagingUnit.local_to_host.version_mapper_a_to_b_rec	PVBusMapper
CombinedMessagingUnit.local_to_host.version_mapper_a_to_b_snd	PVBusMapper
CombinedMessagingUnit.local_to_host_rcv_log	PVBusLogger
CombinedMessagingUnit.local_to_host_rcv_log.mapper	PVBusMapper
CombinedMessagingUnit.local_to_host_snd_log	PVBusLogger
CombinedMessagingUnit.local_to_host_snd_log.mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-617: CombinedMessagingUnit MTI instances**

InstanceName	ComponentName
CombinedMessagingUnit.host_to_local.a_to_b_v2	MessageHandlingUnitV2
CombinedMessagingUnit.host_to_local.a_to_b_v3	MessageHandlingUnitV3
CombinedMessagingUnit.host_to_local.version_mapper_a_to_b_rec	PVBusMapper
CombinedMessagingUnit.host_to_local.version_mapper_a_to_b_snd	PVBusMapper
CombinedMessagingUnit.host_to_local_rcv_log	PVBusLogger
CombinedMessagingUnit.host_to_local_rcv_log.mapper	PVBusMapper
CombinedMessagingUnit.host_to_local_snd_log	PVBusLogger
CombinedMessagingUnit.host_to_local_snd_log.mapper	PVBusMapper
CombinedMessagingUnit.irq_rcv_combined_host_log	SignalLogger
CombinedMessagingUnit.irq_rcv_combined_local_log	SignalLogger
CombinedMessagingUnit.irq_snd_combined_host_log	SignalLogger
CombinedMessagingUnit.irq_snd_combined_local_log	SignalLogger
CombinedMessagingUnit.local_to_host.a_to_b_v2	MessageHandlingUnitV2
CombinedMessagingUnit.local_to_host.a_to_b_v3	MessageHandlingUnitV3

InstanceName	ComponentName
CombinedMessagingUnit.local_to_host.version_mapper_a_to_b_rec	PVBusMapper
CombinedMessagingUnit.local_to_host.version_mapper_a_to_b_snd	PVBusMapper
CombinedMessagingUnit.local_to_host_rcv_log	PVBusLogger
CombinedMessagingUnit.local_to_host_rcv_log.mapper	PVBusMapper
CombinedMessagingUnit.local_to_host_snd_log	PVBusLogger
CombinedMessagingUnit.local_to_host_snd_log.mapper	PVBusMapper

## Ports for CombinedMessagingUnit

**Table 3-618: Ports**

Name	Protocol	Type	Description
irq_rcv_combined_host	Signal	Master	-
irq_rcv_combined_local	Signal	Master	-
irq_snd_combined_host	Signal	Master	-
irq_snd_combined_local	Signal	Master	-
pvbus_s_rcv_host	PVBus	Slave	-
pvbus_s_rcv_local	PVBus	Slave	-
pvbus_s_snd_host	PVBus	Slave	-
pvbus_s_snd_local	PVBus	Slave	-
reset_in	Signal	Slave	-

## Parameters for CombinedMessagingUnit

### NUM\_DB\_CH

#### Type

int

#### Default value

0x1

#### Description

Number of doorbell channels.

### NUM\_FAST\_CH

#### Type

int

#### Default value

0x1

#### Description

Number of fast channels.

**diagnostics****Type**

int

**Default value**

0x2

**Description**

Diagnostics 0==FATAL\_ERROR -&gt; 4==DEBUG.

**fast\_ch\_group\_int\_enable****Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**fast\_ch\_n\_per\_group****Type**

int

**Default value**

0x1

**Description**

Fast Channel num channels per group, default=1.

**fast\_ch\_num\_groups****Type**

int

**Default value**

0x1

**Description**

Fast Channel num of groups, default=1.

**fast\_ch\_word\_size****Type**

int

**Default value**

0x20

**Description**

Fast Channel word size 32bit or 64bit, default=32.

**host\_to\_local.a\_to\_b\_v3.NUM\_FIFO\_CH****Type**

int

**Default value**

0x1

**Description**

Number of FIFO Channels, default=1.

**host\_to\_local.a\_to\_b\_v3.auto\_op\_full****Type**

bool

**Default value**

0x0

**Description**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

**host\_to\_local.a\_to\_b\_v3.fast\_ch\_group\_int\_enable****Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**host\_to\_local.a\_to\_b\_v3.fifo\_depth****Type**

int

**Default value**

0x4

**Description**

Depth of the FIFO = fifo\_depth + 1.

**host\_to\_local.a\_to\_b\_v3.m16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 16 bit access support to FIFO registers.



**host\_to\_local.a\_to\_b\_v3.m32ba\_spt****Type**

bool

**Default value**

0x1

**Description**

Mailbox 32 bit access support to FIFO registers.

**host\_to\_local.a\_to\_b\_v3.m64ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 64 bit access support to FIFO registers.

**host\_to\_local.a\_to\_b\_v3.m8ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 8 bit access support to FIFO registers.

**host\_to\_local.a\_to\_b\_v3.monolithic****Type**

bool

**Default value**

0x1

**Description**

Monolithic or Distributed MHU - default: monolithic(true).

**host\_to\_local.a\_to\_b\_v3.p16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 16 bit access support to FIFO registers.

**host\_to\_local.a\_to\_b\_v3.p32ba\_spt****Type**

bool

**Default value**

0x1

**Description**

Postbox 32 bit access support to FIFO registers.

**host\_to\_local.a\_to\_b\_v3.p64ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 64 bit access support to FIFO registers.

**host\_to\_local.a\_to\_b\_v3.p8ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 8 bit access support to FIFO registers.

**host\_to\_local.fast\_ch\_group\_int\_enable****Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**host\_to\_local\_rcv\_log.trace\_snoops****Type**

bool

**Default value**

0x0

**Description**

Enable tracing of ACE snoop requests.

**host\_to\_local\_snd\_log.trace\_snoops****Type**

bool

**Default value**

0x0

**Description**

Enable tracing of ACE snoop requests.

**irq\_rcv\_combined\_host\_log.forward\_signal****Type**

bool

**Default value**

0x1

**Description**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

**irq\_rcv\_combined\_local\_log.forward\_signal****Type**

bool

**Default value**

0x1

**Description**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

**irq\_snd\_combined\_host\_log.forward\_signal****Type**

bool

**Default value**

0x1

**Description**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

**irq\_snd\_combined\_local\_log.forward\_signal****Type**

bool

**Default value**

0x1

**Description**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

**local\_to\_host.a\_to\_b\_v3.NUM\_FIFO\_CH****Type**

int

**Default value**

0x1

**Description**

Number of FIFO Channels, default=1.

**local\_to\_host.a\_to\_b\_v3.auto\_op\_full****Type**

bool

**Default value**

0x0

**Description**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

**local\_to\_host.a\_to\_b\_v3.fast\_ch\_group\_int\_enable****Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**local\_to\_host.a\_to\_b\_v3.fifo\_depth****Type**

int

**Default value**

0x4

**Description**

Depth of the FIFO = fifo\_depth + 1.

**local\_to\_host.a\_to\_b\_v3.m16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 16 bit access support to FIFO registers.

**local\_to\_host.a\_to\_b\_v3.m32ba\_spt****Type**

bool

**Default value**

0x1

**Description**

Mailbox 32 bit access support to FIFO registers.

**local\_to\_host.a\_to\_b\_v3.m64ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 64 bit access support to FIFO registers.

**local\_to\_host.a\_to\_b\_v3.m8ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 8 bit access support to FIFO registers.

**local\_to\_host.a\_to\_b\_v3.monolithic****Type**

bool

**Default value**

0x1

**Description**

Monolithic or Distributed MHU - default: monolithic(true).

**local\_to\_host.a\_to\_b\_v3.p16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 16 bit access support to FIFO registers.

**local\_to\_host.a\_to\_b\_v3.p32ba\_spt****Type**

bool

**Default value**

0x1

**Description**

Postbox 32 bit access support to FIFO registers.

**local\_to\_host.a\_to\_b\_v3.p64ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 64 bit access support to FIFO registers.

**local\_to\_host.a\_to\_b\_v3.p8ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 8 bit access support to FIFO registers.

**local\_to\_host.fast\_ch\_group\_int\_enable****Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**local\_to\_host\_rcv\_log.trace\_snoops****Type**

bool

**Default value**

0x0

**Description**

Enable tracing of ACE snoop requests.

**local\_to\_host\_snd\_log.trace\_snoops****Type**

bool

**Default value**

0x0

**Description**

Enable tracing of ACE snoop requests.

**major\_version****Type**

int

**Default value**

0x2

**Description**

MHU major version (default=2).

**mhu\_arch\_beta01****Type**

bool

**Default value**

0x0

**Description**

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

**minor\_version****Type**

int

**Default value**

0x1

**Description**

MHU minor version (default=1).

**product\_id****Type**

int

**Default value**

0x0

**Description**

MHU part number.

### 3.7.9 CombinedMessagingUnitAE

CMU AE - Combined MHU320AE monolithic block. This model is written in LISA+.

#### Iris and MTI instances for CombinedMessagingUnitAE

This model has the following Iris instances:

**Table 3-619: CombinedMessagingUnitAE Iris instances**

InstanceName	ComponentName
CombinedMessagingUnitAE	CombinedMessagingUnitAE
CombinedMessagingUnitAE.host_to_local	MHU320AE
CombinedMessagingUnitAE.host_to_local.MHU320AE FMU	mhu320ae_fmu
CombinedMessagingUnitAE.host_to_local_rcv_log	PVBusLogger
CombinedMessagingUnitAE.host_to_local_rcv_log.mapper	PVBusMapper
CombinedMessagingUnitAE.host_to_local_snd_log	PVBusLogger
CombinedMessagingUnitAE.host_to_local_snd_log.mapper	PVBusMapper
CombinedMessagingUnitAE.irq_rcv_combined_host_log	SignalLogger
CombinedMessagingUnitAE.irq_rcv_combined_local_log	SignalLogger
CombinedMessagingUnitAE.irq_snd_combined_host_log	SignalLogger
CombinedMessagingUnitAE.irq_snd_combined_local_log	SignalLogger
CombinedMessagingUnitAE.local_to_host	MHU320AE
CombinedMessagingUnitAE.local_to_host.MHU320AE FMU	mhu320ae_fmu
CombinedMessagingUnitAE.local_to_host_rcv_log	PVBusLogger
CombinedMessagingUnitAE.local_to_host_rcv_log.mapper	PVBusMapper
CombinedMessagingUnitAE.local_to_host_snd_log	PVBusLogger
CombinedMessagingUnitAE.local_to_host_snd_log.mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-620: CombinedMessagingUnitAE MTI instances**

InstanceName	ComponentName
CombinedMessagingUnitAE.host_to_local	MessageHandlingUnitV3
CombinedMessagingUnitAE.host_to_local_rcv_log	PVBusLogger
CombinedMessagingUnitAE.host_to_local_rcv_log.mapper	PVBusMapper
CombinedMessagingUnitAE.host_to_local_snd_log	PVBusLogger
CombinedMessagingUnitAE.host_to_local_snd_log.mapper	PVBusMapper



InstanceName	ComponentName
CombinedMessagingUnitAE.irq_rcv_combined_host_log	SignalLogger
CombinedMessagingUnitAE.irq_rcv_combined_local_log	SignalLogger
CombinedMessagingUnitAE.irq_snd_combined_host_log	SignalLogger
CombinedMessagingUnitAE.irq_snd_combined_local_log	SignalLogger
CombinedMessagingUnitAE.local_to_host	MessageHandlingUnitV3
CombinedMessagingUnitAE.local_to_host_rcv_log	PVBusLogger
CombinedMessagingUnitAE.local_to_host_rcv_log.mapper	PVBusMapper
CombinedMessagingUnitAE.local_to_host_snd_log	PVBusLogger
CombinedMessagingUnitAE.local_to_host_snd_log.mapper	PVBusMapper

## Ports for CombinedMessagingUnitAE

**Table 3-621: Ports**

Name	Protocol	Type	Description
fmu_cri_out_host	Signal	Master	-
fmu_cri_out_local	Signal	Master	-
fmu_eri_out_host	Signal	Master	-
fmu_eri_out_local	Signal	Master	-
irq_rcv_combined_host	Signal	Master	-
irq_rcv_combined_local	Signal	Master	-
irq_snd_combined_host	Signal	Master	-
irq_snd_combined_local	Signal	Master	-
pvbus_s_rcv_host	PVBus	Slave	-
pvbus_s_rcv_local	PVBus	Slave	-
pvbus_s_snd_fmu_host	PVBus	Slave	-
pvbus_s_snd_fmu_local	PVBus	Slave	-
pvbus_s_snd_host	PVBus	Slave	-
pvbus_s_snd_local	PVBus	Slave	-
reset_in	Signal	Slave	-

## Parameters for CombinedMessagingUnitAE

### NUM\_DB\_CH

#### Type

int

#### Default value

0x1

#### Description

Number of doorbell channels.

**NUM\_FAST\_CH****Type**

int

**Default value**

0x1

**Description**

Number of fast channels.

**diagnostics****Type**

int

**Default value**

0x2

**Description**

Diagnostics 0==FATAL\_ERROR -&gt; 4==DEBUG.

**fast\_ch\_group\_int\_enable****Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**fast\_ch\_n\_per\_group****Type**

int

**Default value**

0x1

**Description**

Fast Channel num channels per group, default=1.

**fast\_ch\_num\_groups****Type**

int

**Default value**

0x1

**Description**

Fast Channel num of groups, default=1.

**fast\_ch\_word\_size****Type**

int

**Default value**

0x20

**Description**

Fast Channel word size 32bit or 64bit, default=32.

**host\_to\_local.NUM\_FIFO\_CH****Type**

int

**Default value**

0x1

**Description**

Number of FIFO Channels, default=1.

**host\_to\_local.auto\_op\_full****Type**

bool

**Default value**

0x0

**Description**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

**host\_to\_local.fast\_ch\_group\_int\_enable****Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**host\_to\_local.fifo\_depth****Type**

int

**Default value**

0x4

**Description**

Depth of the FIFO = fifo\_depth + 1.

**host\_to\_local.fmu\_location****Type**

int

**Default value**

0x2

**Description**

fmu\_location 0-2 (0:SENDER 1:RECEIVER 2:BOTH).

**host\_to\_local.m16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 16 bit access support to FIFO registers.

**host\_to\_local.m32ba\_spt****Type**

bool

**Default value**

0x1

**Description**

Mailbox 32 bit access support to FIFO registers.

**host\_to\_local.m64ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 64 bit access support to FIFO registers.

**host\_to\_local.m8ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 8 bit access support to FIFO registers.

**host\_to\_local.monolithic****Type**

bool

**Default value**

0x1

**Description**

Monolithic or Distributed MHU - default: monolithic(true).

**host\_to\_local.p16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 16 bit access support to FIFO registers.

**host\_to\_local.p32ba\_spt****Type**

bool

**Default value**

0x1

**Description**

Postbox 32 bit access support to FIFO registers.

**host\_to\_local.p64ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 64 bit access support to FIFO registers.

**host\_to\_local.p8ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 8 bit access support to FIFO registers.

**host\_to\_local\_rcv\_log.trace\_snoops****Type**

bool

**Default value**

0x0

**Description**

Enable tracing of ACE snoop requests.

**host\_to\_local\_snd\_log.trace\_snoops****Type**

bool

**Default value**

0x0

**Description**

Enable tracing of ACE snoop requests.

**irq\_rcv\_combined\_host\_log.forward\_signal****Type**

bool

**Default value**

0x1

**Description**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

**irq\_rcv\_combined\_local\_log.forward\_signal****Type**

bool

**Default value**

0x1

**Description**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

**irq\_snd\_combined\_host\_log.forward\_signal****Type**

bool

**Default value**

0x1

**Description**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

**irq\_snd\_combined\_local\_log.forward\_signal****Type**

bool

**Default value**

0x1

**Description**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

**local\_to\_host.NUM\_FIFO\_CH****Type**

int

**Default value**

0x1

**Description**

Number of FIFO Channels, default=1.

**local\_to\_host.auto\_op\_full****Type**

bool

**Default value**

0x0

**Description**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

**local\_to\_host.fast\_ch\_group\_int\_enable****Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**local\_to\_host.fifo\_depth****Type**

int

**Default value**

0x4

**Description**

Depth of the FIFO = ffo\_depth + 1.

**local\_to\_host.fmu\_location****Type**

int

**Default value**

0x2

**Description**

fmu\_location 0-2 (0:SENDER 1:RECEIVER 2:BOTH).

**local\_to\_host.m16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 16 bit access support to FIFO registers.

**local\_to\_host.m32ba\_spt****Type**

bool

**Default value**

0x1

**Description**

Mailbox 32 bit access support to FIFO registers.

**local\_to\_host.m64ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 64 bit access support to FIFO registers.

**local\_to\_host.m8ba\_spt****Type**

bool



**Default value**

0x0

**Description**

Mailbox 8 bit access support to FIFO registers.

**local\_to\_host.monolithic****Type**

bool

**Default value**

0x1

**Description**

Monolithic or Distributed MHU - default: monolithic(true).

**local\_to\_host.p16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 16 bit access support to FIFO registers.

**local\_to\_host.p32ba\_spt****Type**

bool

**Default value**

0x1

**Description**

Postbox 32 bit access support to FIFO registers.

**local\_to\_host.p64ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 64 bit access support to FIFO registers.

**local\_to\_host.p8ba\_spt****Type**

bool

**Default value**  
0x0

**Description**  
Postbox 8 bit access support to FIFO registers.

**local\_to\_host\_rcv\_log.trace\_snoops**

**Type**  
bool

**Default value**  
0x0

**Description**  
Enable tracing of ACE snoop requests.

**local\_to\_host\_snd\_log.trace\_snoops**

**Type**  
bool

**Default value**  
0x0

**Description**  
Enable tracing of ACE snoop requests.

3.7.10 DebugAccessPort

This model is written in C++.

Iris and MTI instances for DebugAccessPort

This model has the following Iris instances:

Table 3-622: DebugAccessPort Iris instances

InstanceName	ComponentName
DebugAccessPort	dap

Ports for DebugAccessPort

Table 3-623: Ports

Name	Protocol	Type	Description
ap_pvbus_m[2]	PVBus	Master	-
clock	ClockSignal	Slave	-
paddrdbg31	Signal	Master	-

## Parameters for DebugAccessPort

### **ap0\_has\_debug\_rom**

#### Type

bool

#### Default value

0x0

#### Description

Whether AP0 has a Debug ROM.

### **ap0\_rom\_base\_address**

#### Type

int

#### Default value

0x0

#### Description

ROM base address for AP 0.

### **ap0\_set\_paddrdbg31**

#### Type

bool

#### Default value

0x0

#### Description

Set paddrdbg31 signal during accesses on AP0.

### **ap1\_has\_debug\_rom**

#### Type

bool

#### Default value

0x0

#### Description

Whether AP1 has a Debug ROM.

### **ap1\_rom\_base\_address**

#### Type

int

#### Default value

0x0

**Description**

ROM base address for AP 1.

**ap1\_set\_paddrdbg31**

**Type**

bool

**Default value**

0x0

**Description**

Set paddrdbg31 signal during accesses on AP1.

3.7.11 DebugROM

This model is written in C++.

**Changes in 11.29.19**

Parameters added:

- ROMDEVID
- ROMPIDR
- ROMPRIDR0

**About DebugRom**

Debug ROM complying to an ADiv5-like interface

**Iris and MTI instances for DebugROM**

This model has the following Iris instances:

Table 3-624: DebugROM Iris instances

InstanceName	ComponentName
DebugROM	debug_rom

Ports for DebugROM

Table 3-625: Ports

Name	Protocol	Type	Description
paddrdbg31	Signal	Master	Signal port for paddrdbg to recognize an external access
pvbuss_s	PVBus	Slave	Bus slave port for accessing registers

## Parameters for DebugROM

### **ROMDEVID**

**Type**

int

**Default value**

0x0

**Description**

Initial value of Debug Rom Device Identification Register.

### **ROMPIDR**

**Type**

int

**Default value**

0x4000bb000

**Description**

Initial value of Debug Rom Peripheral Identification Register. Setting this parameter would make manufacturer\_revision\_number, customer\_modified, revision, and part\_number parameters take no effect.

### **ROMPRIDR0**

**Type**

int

**Default value**

0x1

**Description**

Initial value of Debug ROM Power RequestID Register.

### **customer\_modified**

**Type**

int

**Default value**

0x0

**Description**

### **entry\_0**

**Type**

int

**Default value**

0x0

**Description**

Offset of component 0.

**entry\_1****Type**

int

**Default value**

0x0

**Description**

Offset of component 1.

**entry\_10****Type**

int

**Default value**

0x0

**Description**

Offset of component 10.

**entry\_11****Type**

int

**Default value**

0x0

**Description**

Offset of component 11.

**entry\_12****Type**

int

**Default value**

0x0

**Description**

Offset of component 12.

**entry\_13****Type**

int

**Default value**

0x0

**Description**

Offset of component 13.

**entry\_14****Type**

int

**Default value**

0x0

**Description**

Offset of component 14.

**entry\_15****Type**

int

**Default value**

0x0

**Description**

Offset of component 15.

**entry\_16****Type**

int

**Default value**

0x0

**Description**

Offset of component 16.

**entry\_17****Type**

int

**Default value**

0x0

**Description**

Offset of component 17.

**entry\_18****Type**

int

**Default value**

0x0

**Description**

Offset of component 18.

**entry\_19****Type**

int

**Default value**

0x0

**Description**

Offset of component 19.

**entry\_2****Type**

int

**Default value**

0x0

**Description**

Offset of component 2.

**entry\_20****Type**

int

**Default value**

0x0

**Description**

Offset of component 20.

**entry\_21****Type**

int

**Default value**

0x0

**Description**

Offset of component 21.

**entry\_22****Type**

int

**Default value**

0x0



**Description**

Offset of component 22.

**entry\_23****Type**

int

**Default value**

0x0

**Description**

Offset of component 23.

**entry\_24****Type**

int

**Default value**

0x0

**Description**

Offset of component 24.

**entry\_25****Type**

int

**Default value**

0x0

**Description**

Offset of component 25.

**entry\_26****Type**

int

**Default value**

0x0

**Description**

Offset of component 26.

**entry\_27****Type**

int

**Default value**

0x0

**Description**

Offset of component 27.

**entry\_28****Type**

int

**Default value**

0x0

**Description**

Offset of component 28.

**entry\_29****Type**

int

**Default value**

0x0

**Description**

Offset of component 29.

**entry\_3****Type**

int

**Default value**

0x0

**Description**

Offset of component 3.

**entry\_30****Type**

int

**Default value**

0x0

**Description**

Offset of component 30.

**entry\_31****Type**

int

**Default value**

0x0

**Description**

Offset of component 31.

**entry\_32****Type**

int

**Default value**

0x0

**Description**

Offset of component 32.

**entry\_33****Type**

int

**Default value**

0x0

**Description**

Offset of component 33.

**entry\_34****Type**

int

**Default value**

0x0

**Description**

Offset of component 34.

**entry\_35****Type**

int

**Default value**

0x0

**Description**

Offset of component 35.

**entry\_36****Type**

int

**Default value**

0x0

**Description**

Offset of component 36.

**entry\_37****Type**

int

**Default value**

0x0

**Description**

Offset of component 37.

**entry\_38****Type**

int

**Default value**

0x0

**Description**

Offset of component 38.

**entry\_39****Type**

int

**Default value**

0x0

**Description**

Offset of component 39.

**entry\_4****Type**

int

**Default value**

0x0

**Description**

Offset of component 4.

**entry\_40****Type**

int

**Default value**

0x0

**Description**

Offset of component 40.

**entry\_41****Type**

int

**Default value**

0x0

**Description**

Offset of component 41.

**entry\_42****Type**

int

**Default value**

0x0

**Description**

Offset of component 42.

**entry\_43****Type**

int

**Default value**

0x0

**Description**

Offset of component 43.

**entry\_44****Type**

int

**Default value**

0x0

**Description**

Offset of component 44.

**entry\_45****Type**

int

**Default value**

0x0

**Description**

Offset of component 45.

**entry\_46****Type**

int

**Default value**

0x0

**Description**

Offset of component 46.

**entry\_47****Type**

int

**Default value**

0x0

**Description**

Offset of component 47.

**entry\_48****Type**

int

**Default value**

0x0

**Description**

Offset of component 48.

**entry\_49****Type**

int

**Default value**

0x0

**Description**

Offset of component 49.

**entry\_5****Type**

int

**Default value**

0x0

**Description**

Offset of component 5.

**entry\_50****Type**

int

**Default value**

0x0

**Description**

Offset of component 50.

**entry\_51****Type**

int

**Default value**

0x0

**Description**

Offset of component 51.

**entry\_52****Type**

int

**Default value**

0x0

**Description**

Offset of component 52.

**entry\_53****Type**

int

**Default value**

0x0

**Description**

Offset of component 53.

**entry\_54****Type**

int

**Default value**

0x0

**Description**

Offset of component 54.

**entry\_55****Type**

int

**Default value**

0x0

**Description**

Offset of component 55.

**entry\_56****Type**

int

**Default value**

0x0

**Description**

Offset of component 56.

**entry\_57****Type**

int

**Default value**

0x0

**Description**

Offset of component 57.

**entry\_58****Type**

int

**Default value**

0x0

**Description**

Offset of component 58.

**entry\_59****Type**

int

**Default value**

0x0



**Description**

Offset of component 59.

**entry\_6****Type**

int

**Default value**

0x0

**Description**

Offset of component 6.

**entry\_60****Type**

int

**Default value**

0x0

**Description**

Offset of component 60.

**entry\_61****Type**

int

**Default value**

0x0

**Description**

Offset of component 61.

**entry\_62****Type**

int

**Default value**

0x0

**Description**

Offset of component 62.

**entry\_63****Type**

int

**Default value**

0x0

**Description**

Offset of component 63.

**entry\_7****Type**

int

**Default value**

0x0

**Description**

Offset of component 7.

**entry\_8****Type**

int

**Default value**

0x0

**Description**

Offset of component 8.

**entry\_9****Type**

int

**Default value**

0x0

**Description**

Offset of component 9.

**manufacturer\_revision\_number****Type**

int

**Default value**

0x0

**Description****part\_number****Type**

int

**Default value**

0x0

**Description****revision****Type**

int

**Default value**

0x0

**Description**

### 3.7.12 DualClusterSystemConfigurationBlock

Dual Cluster System Configuration Block. This model is written in LISA+.

#### Iris and MTI instances for DualClusterSystemConfigurationBlock

This model has the following Iris instances:

**Table 3-626: DualClusterSystemConfigurationBlock Iris instances**

InstanceName	ComponentName
DualClusterSystemConfigurationBlock	DualClusterSystemConfigurationBlock
DualClusterSystemConfigurationBlock.pvbusslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-627: DualClusterSystemConfigurationBlock MTI instances**

InstanceName	ComponentName
DualClusterSystemConfigurationBlock	DualClusterSystemConfigurationBlock
DualClusterSystemConfigurationBlock.pvbusslave	PVBusSlave

#### Ports for DualClusterSystemConfigurationBlock

**Table 3-628: Ports**

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	-
cluster0_cfgend[4]	Signal	Master	-
cluster0_cfgte[4]	Signal	Master	-
cluster0_clusterid	Value	Master	-
cluster0_corereset[4]	Signal	Master	-
cluster0_cpuporeset[4]	Signal	Master	-
cluster0_cxreset[4]	Signal	Master	-
cluster0_eventi	Signal	Peer	-
cluster0_evento	Signal	Peer	-
cluster0_iminlen	Signal	Master	-

Name	Protocol	Type	Description
cluster0_l2reset	Signal	Master	-
cluster0_standbywfi[4]	Signal	Slave	-
cluster0_vinithi[4]	Signal	Master	-
cluster1_cfgend[4]	Signal	Master	-
cluster1_clusterid	Value	Master	-
cluster1_corereset[4]	Signal	Master	-
cluster1_cpuporeset[4]	Signal	Master	-
cluster1_eventi	Signal	Peer	-
cluster1_evento	Signal	Peer	-
cluster1_scureset	Signal	Master	-
cluster1_standbywfi[4]	Signal	Slave	-
cluster1_teinit[4]	Signal	Master	-
cluster1_vinithi[4]	Signal	Master	-
daughter_leds_state	ValueState	Master	-
daughter_user_switches	ValueState	Master	-
intgen[128]	Signal	Master	-
periphbase	Value_64	Master	-
periphbase_32	Value	Master	-
pvbus	PVBus	Slave	-
system_reset	Signal	Master	-
vgic_configuration_port	v7_VGIC_Configuration_Protocol	Master	-

## Parameters for DualClusterSystemConfigurationBlock

### CFG\_ACTIVECLUSTER

#### Type

int

#### Default value

0x1

#### Description

Select which cluster will come out of reset coming out of power-on: bit[0] for primary cluster (Cortex-A15), bit[1] for secondary cluster (Cortex-A7). Value 0 is not allowed as it will hold both clusters in reset indefinitely!.

### Cluster0IdOnPOReset

#### Type

int

#### Default value

0x0

**Description**

ClusterId for primary cluster (Cortex-A15) on power-on reset.

**Cluster1IdOnPOReset****Type**

int

**Default value**

0x1

**Description**

ClusterId for secondary cluster (Cortex-A7) on power-on reset.

**DCSCB\_PERIPHBASE****Type**

int

**Default value**

0x1e000000

**Description**

PERIPHBASE.

**DCS\_AID****Type**

int

**Default value**

0x0

**Description**

DCS\_AID is the Auxiliary ID Register.

**DCS\_ID****Type**

int

**Default value**

0x41120000

**Description**

The value returned by the DCS\_ID register.

**DCS\_ID\_BUILD\_NUMBER****Type**

int

**Default value**

0x1

**Description**

DCS\_ID build number.

**DCS\_LEDS****Type**

int

**Default value**

0x0

**Description**

DCS\_LEDS represents eight LEDs on the board that form an 8-bit value that can be r/w from the Dual Cluster System Configuration Block.

**DCS\_SW****Type**

int

**Default value**

0x0

**Description**

DCS\_SW represents eight switches on the board that form an 8-bit value that can be read from the Dual Cluster System Configuration Block.

**FlipVGICWiringForCluster0AndCluster1****Type**

bool

**Default value**

0x0

**Description**

Flip the VGIC wiring round for cluster0 and cluster1. With this false, then cpu0 of cluster0 is cpu interface 0 on the VGIC. If this is true then cpu0 of cluster1 becomes cpu interface 0 on the VGIC.

**INTGEN\_INTS****Type**

int

**Default value**

0x3

**Description**

Number of custom IRQs controlled by interrupt generator is  $\text{INTGEN\_INTS} * 32 + 32$ .

**NumberOfCoresInCluster0****Type**

int

**Default value**

0x2

**Description**

The number of cores in the primary cluster.

**NumberOfCoresInCluster1****Type**

int

**Default value**

0x2

**Description**

The number of cores in the secondary cluster.

**ResetValueOfDaughterUserSwitches****Type**

int

**Default value**

0x0

**Description**

Reset value of the user switches on the daughterboard.

**stop\_on\_sequence\_id****Type**

int

**Default value**

0x0

**Description**

If non-zero the sequence\_id of the SW trace mechanism on which to halt the simulator.

### 3.7.13 DummyAPB

DummyAPB. This model is written in LISA+.

**About DummyAPB**

Use this dummy **RAZ/WI** APB device component to ensure that software does not receive aborts for accesses to devices that should be part of the system, but are not modeled.

For validation purposes it is useful to have dummy devices that are mostly **RAZ/WI** but return the correct value when you read ID registers. You can do that with this component in the following ways:

- Specify `periphid_24` for peripherals that follow the Arm pattern of having 12 ID registers at the top of an APB frame. For example:

```
periphid_24="04000000c2b00b000df005b1"
```

You also must set `periph_framesize` to 4 or 64, depending on whether the peripheral has its registers in a 4 KB or 64 KB frame.

- Give a space-separated list of offset:value pairs in the `periphid_generic` parameter to define read-only values from particular offsets. For example:

```
periphid_generic="000:02468ace 1fc:13579bdf"
```

The number of hex digits used to specify the address is used to define the width of the address mask used. For example, `bc:02468ace` returns `02468ace` at reads from any address ending `bc`.

- Give a space-separated list of offset:default-value pairs in the `ram_generic` parameter to construct RAM. That is, the register at the relevant offset returns the default-value, but if changed, it returns the value that it is changed to.

Iris and MTI instances for DummyAPB

This model has the following Iris instances:

Table 3-629: DummyAPB Iris instances

InstanceName	ComponentName
DummyAPB	DummyAPB
DummyAPB.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Table 3-630: DummyAPB MTI instances

InstanceName	ComponentName
DummyAPB.pvbusslave	PVBusSlave

Ports for DummyAPB

Table 3-631: Ports

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	Bus slave interface.



## Parameters for DummyAPB

### **fail**

#### Type

bool

#### Default value

0x0

#### Description

Abort all accesses.

### **failmsg**

#### Type

string

#### Default value

""

#### Description

String to print when 'fail'=true and access occurred.

### **periph\_framesize**

#### Type

int

#### Default value

0xffffffffffffffff

#### Description

Size of frame (4/64, indicating if ID is at xFD0 or xFFD0).

### **periphid\_24**

#### Type

string

#### Default value

""

#### Description

24 hex digits for the 12 bytes of peripheral ID.

### **periphid\_generic**

#### Type

string

#### Default value

""

**Description**

Set of space-separated offset:value pairs for dwords of ID.

**ram\_generic****Type**

string

**Default value**

""

**Description**

Set of space-separated offset:default pairs for writable dwords.

**warn\_once****Type**

bool

**Default value**

0x1

**Description**

Warn once for the invalid read and write access.

### 3.7.14 ElfLoader

ELF loader component. This model is written in LISA+.

**About ElfLoader**

ElfLoader provides an alternative method of loading ELF files into the system. It can load files in either of the following formats, or in gzip-compressed versions of them:

- ELF
- Motorola S-Record

**Load file format**

ElfLoader optionally uses a load file to load code and data in an ELF file to any address in the physical address space before the simulation starts.

Specify the ELF file to load using the `elf` parameter and the load file using the `lfile` parameter. The load file contains an entry for each section in the ELF file. An entry has the following format:

```
<Section name>, <VA>, <PA>, <Offset>, <Size>, <PAS>, <MECID>
```

Where:

**Section name**

The name of a segment of continuous code and data to be preloaded. It must match a segment name in the ELF file. It can only contain alphanumeric characters a - z, A - Z, 0 - 9, and underscores (\_).

**VA**

The Virtual Address of the code and data in hexadecimal. This is the same as the address of the segment in the ELF file.

**PA**

The Physical Address in hexadecimal to which the code and data will be preloaded in memory.

**Offset**

Must be set to 0x0. Reserved for future use.

**Size**

The size of the data to be preloaded at the Physical Address (PA) in hexadecimal. It is the sum of the sizes of the individual ELF sections belonging to the same segment. It should match the `memsz` field of the corresponding segment in the ELF file.

**PAS**

The Physical Address Space to which the code and data will be preloaded. The possible values are:

**S**

Secure PAS

**NS**

Non-secure PAS

**RL**

Realm PAS

**RT**

Root PAS

**MECID**

Memory Encryption Context ID (optional). It can be specified for any PAS.

The following rules apply to the load file:

- The load file definition starts with the line:

```
** ELF_SECTION_RELOC_START **
```

and ends with the line:

```
** ELF_SECTION_RELOC_END **
```

- Lines within the load file definition that begin with # and all lines outside the definition are treated as comments.

- The hexadecimal values in the VA and PA fields must begin with 0x.
- Fields are separated by a comma and all whitespace characters are ignored.
- Blank lines within the load file definition are ignored.

### Example load file

This example load file shows a segment of size 15 KB with a start address in VA space of 0x500000. It is preloaded to address 0x600000 in non-secure physical memory:

```
** ELF_SECTION_RELOC_START **
# Section relocation for Code Segment 1
PRELOAD_TEST_1, 0x500000, 0x600000, 0x0, 0x3C00, NS
** ELF_SECTION_RELOC_END **
```

### ns\_copy, realm\_copy, and root\_copy parameters

The boolean parameters `ns_copy`, `realm_copy`, and `root_copy` are alternatives to `lfile`. If `lfile` is also specified, it takes precedence over them.

If you enable any of these boolean parameters, the ELF file is loaded to both Secure PAS and to the PAS in the parameter name. So, for example, enabling all three parameters loads the ELF file to all four PASes.

If you do not specify `lfile` and do not enable any of the `<PAS>_copy` parameters, the file is loaded to Secure PAS only.

Use `lfile` instead of the `<PAS>_copy` parameters if you want control over:

- Where different segments in the ELF file are loaded to
- Whether to load the ELF file to Secure PAS
- The PA to load to and the MECID to use

### Iris and MTI instances for ElfLoader

This model has the following Iris instances:

**Table 3-632: ElfLoader Iris instances**

InstanceName	ComponentName
ElfLoader	ElfLoader
ElfLoader.pvbus_busmaster	PVBusMaster

This model has the following MTI trace components:

**Table 3-633: ElfLoader MTI instances**

InstanceName	ComponentName
ElfLoader.pvbus_busmaster	PVBusMaster

Ports for ElfLoader

Table 3-634: Ports

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Master port for all memory accesses.
start_address	Value_64	Master	Provides a value reflecting the entry point of the last ELF image to be loaded.

Parameters for ElfLoader

**elf**

Type

string

Default value

""

Description

ELF file.

**impdef\_copy**

Type

bool

Default value

0x0

Description

DEPRECATED: Use realm\_copy or root\_copy parameters. load ELF file to implementation defined memory spaces, if load file is not specified.

**lfile**

Type

string

Default value

""

Description

load file for large address mapping.

**ns\_copy**

Type

bool

Default value

0x1

Description

copy whole file to NS memory space.

**output\_attributes\_parameter\_of\_core**

**Type**  
string

**Default value**  
"ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_SP[0],  
ExtendedID[37]=MPAM\_SP[1], UserFlags[31:16]=IMPDEF2"

**Description**  
Encoding of various attributes on the bus.

**realm\_copy**

**Type**  
bool

**Default value**  
0x0

**Description**  
load ELF file to REALM memory spaces, if load file is not specified.

**root\_copy**

**Type**  
bool

**Default value**  
0x0

**Description**  
load ELF file to ROOT memory spaces, if load file is not specified.

3.7.15 FlashLoader

A device that can preload a gzipped image into flash at startup. This model is written in C++.

About FlashLoader

This component complements the IntelStrataFlashJ3 component by providing a means to initialize the contents of up to four Flash components in sequence from a single host flash image file.

Iris and MTI instances for FlashLoader

This model has the following Iris instances:

Table 3-635: FlashLoader Iris instances

InstanceName	ComponentName
FlashLoader	FlashLoader

## Ports for FlashLoader

**Table 3-636: Ports**

Name	Protocol	Type	Description
flash_device0	FlashLoaderPort	Master	Used to program a flash device.
flash_device1	FlashLoaderPort	Master	Used to program a flash device.
flash_device2	FlashLoaderPort	Master	Used to program a flash device.
flash_device3	FlashLoaderPort	Master	Used to program a flash device.
warm_reset	Signal	Slave	Reset signal from external master.

## Parameters for FlashLoader

### Diagnostics

#### Type

int

#### Default value

0x0

#### Description

Diagnostics.

### fname

#### Type

string

#### Default value

"(none)"

#### Description

Filename (Default '(none)' means: Do not load any file. An empty string will cause a warning.).

### fnameWrite

#### Type

string

#### Default value

"(none)"

#### Description

FilenameWrite (Default '(none)' means: Do not save any file. An empty string will cause a warning.).

### write\_flash\_after\_reset

#### Type

bool

**Default value**

0x0

**Description**

write\_flash\_after\_reset.

3.7.16 GICv3CommsLogger

Traces GICv3Comms activity. This model is written in LISA+.

**Iris and MTI instances for GICv3CommsLogger**

This model has the following Iris instances:

Table 3-637: GICv3CommsLogger Iris instances

InstanceName	ComponentName
GICv3CommsLogger	GICv3CommsLogger

This model has the following MTI trace components:

Table 3-638: GICv3CommsLogger MTI instances

InstanceName	ComponentName
GICv3CommsLogger	GICv3CommsLogger

**Ports for GICv3CommsLogger**

Table 3-639: Ports

Name	Protocol	Type	Description
to_cpu	GICv3Comms	Master	To connect to CPU.
to_gic	GICv3Comms	Slave	To connect to GIC.

**Parameters for GICv3CommsLogger**

**verbose**

**Type**

bool

**Default value**

0x0

**Description**

Print tracing information to attached debugger in addition to via MTI.



### 3.7.17 GICv3CommsPVBUS

GICv3 Component for conversion between GICv3Comms protocol and PVBUS. This model is written in C++.

#### Iris and MTI instances for GICv3CommsPVBUS

This model has the following Iris instances:

**Table 3-640: GICv3CommsPVBUS Iris instances**

InstanceName	ComponentName
GICv3CommsPVBUS	GICv3CommsPVBUS
GICv3CommsPVBUS.bus_slave	PVBUSSlave

This model has the following MTI trace components:

**Table 3-641: GICv3CommsPVBUS MTI instances**

InstanceName	ComponentName
GICv3CommsPVBUS	GICv3CommsPVBUS
GICv3CommsPVBUS.bus_slave	PVBUSSlave

#### Ports for GICv3CommsPVBUS

**Table 3-642: Ports**

Name	Protocol	Type	Description
axi_master_id_s[256]	Value	Slave	-
distributor_s[256]	GICv3Comms	Slave	-
pvbuss_m	PVBUS	Master	-
pvbuss_s	PVBUS	Slave	-

### 3.7.18 GICv3ProtocolChecker

GICv3 Component for command protocol checking. This model is written in C++.

#### Iris and MTI instances for GICv3ProtocolChecker

This model has the following Iris instances:

**Table 3-643: GICv3ProtocolChecker Iris instances**

InstanceName	ComponentName
GICv3ProtocolChecker	GICv3ProtocolChecker

This model has the following MTI trace components:

**Table 3-644: GICv3ProtocolChecker MTI instances**

InstanceName	ComponentName
GICv3ProtocolChecker	GICv3ProtocolChecker

### Ports for GICv3ProtocolChecker

**Table 3-645: Ports**

Name	Protocol	Type	Description
cpu_comms	GICv3Comms	Master	Master GICv3Comms port.
gicv3_comms	GICv3Comms	Slave	Slave GICv3Comms port.

### Parameters for GICv3ProtocolChecker

#### **cpu\_interface\_id**

##### Type

int

##### Default value

0x0

##### Description

Cpu interface id to which this component is connected.

#### **enable\_protocol\_checking**

##### Type

bool

##### Default value

0x1

##### Description

Enable/disable the protocol checking.

## 3.7.19 GUIPoll

Component providing a real-time periodic callback for GUI refresh. This model is written in C++.

### About GUIPoll

An external subcomponent that encapsulates support for generating a real-time callback signal that can be used to poll the event queue of a visualisation GUI.

The `gui_callback()` method of the callback port is invoked periodically, at approximately the rate determined by the `delay_ms` parameter.



This callback is real-time, not simulation-time. Also, callbacks will continue even while the simulation is paused. Because of this, the client code should not implement a callback behavior that can modify the state of the simulation.

### Ports for GUIPoll

Table 3-646: Ports

Name	Protocol	Type	Description
gui_callback	GUIPollCallback	Master	Sends callback requests to the visualization component.

### Parameters for GUIPoll

**delay\_ms**

Type

int

Default value

0x32

Description

GUI update period in ms.

**has\_gui**

Type

bool

Default value

0x1

Description

GUI is enabled.

## 3.7.20 HostBridge

Host Socket Interface Component. This model is written in C++.

### About HostBridge

This component acts as a networking proxy for target NIC device models, to forward and receive ethernet packets to and from the host. Two kinds of proxy backend are integrated into this component:

- A host TAP/TUN-like network device, which is an ordinary TAP or MacVTap. This is the default.
- User-mode networking, which emulates a built-in IP router and DHCP server to route traffic using the host user-mode socket layer. To enable user-mode networking, set the `userNetworking` parameter to true.

HostBridge requires the following initialization sequence:

```
hostbridge.state.setValue(HostBridge::STATUS);  
hostbridge.state.setValue(HostBridge::S_UP);
```

To enable tracing of user-mode networking, which can help to debug networking issues, set the FASTSIM\_USERNET\_DUMP environment variable to any or all of the following values:

```
arpin,arpout,udpin,udpout,etherin,etherout,ipv4in,ipv4out,  
ipv4fragin,ipv4fragout,tcpin,tcpout,dhcpv4in,dhcpv4out
```

See also

- [Configuring the networking environment for Linux](#)
- [User mode networking](#)

Iris and MTI instances for HostBridge

This model has the following Iris instances:

Table 3-647: HostBridge Iris instances

InstanceName	ComponentName
HostBridge	HostBridge

Ports for HostBridge

Table 3-648: Ports

Name	Protocol	Type	Description
eth	VirtualEthernet	Slave	-
state	ValueState_64	Slave	-

Parameters for HostBridge

**interfaceName**

Type

string

Default value

""

Description

Host Interface.

**userNetOptions**

Type

string

**Default value**

""

**Description**

Control options for UserNet TCP/IP (for internal use only, please do not use).

**userNetPorts**

**Type**

string

**Default value**

""

**Description**

Listening ports to expose in user-mode networking.

**userNetSubnet**

**Type**

string

**Default value**

"172.20.51.0/24"

**Description**

Virtual subnet for user-mode networking.

**userNetworking**

**Type**

bool

**Default value**

0x0

**Description**

Enable user-mode networking.

**3.7.21 HostSerialInterface**

Component which provides access to the host serial interface. This model is written in LISA+.

**Iris and MTI instances for HostSerialInterface**

This model has the following Iris instances:

**Table 3-649: HostSerialInterface Iris instances**

InstanceName	ComponentName
HostSerialInterface	HostSerialInterface

## Ports for HostSerialInterface

**Table 3-650: Ports**

Name	Protocol	Type	Description
SerialData	<a href="#">SerialData</a>	Slave	Serial data connection to export to host machine.

### Parameters for HostSerialInterface

#### **baud\_rate**

##### Type

int

##### Default value

0x0

##### Description

Baud rate override.

#### **device**

##### Type

string

##### Default value

"/dev/ttyS0"

##### Description

HW device to use.

## 3.7.22 IntelStrataFlashJ3

Intel Strata Flash J3 model. This model is written in C++.

### About IntelStrataFlashJ3

This component is an efficient implementation of a NOR flash memory type device, an Intel StrataFlash Memory (J3). For information about Intel StrataFlash Memory (J3), see [Intel Download Center, Intel StrataFlash Memory \(J3\) datasheet](#).

In normal usage, the device acts as Read Only Memory (ROM) whose contents can be determined either by programming using the flashloader port or by using standard flash programming software running on the model, such as the Arm Firmware Suite.

The implementation of this component is approximately that of the Intel part in the VE development board. The component is effectively organized as a bank of two 16-bit Intel Flash components forming a 32-bit component that can be read or programmed in parallel. The component supports all hardware behavior except for:

- Protection register.
- Enhanced configuration register.

- Unique device identifier.
- One time programmable cells.
- Suspend/resume, which is silently ignored.
- Status interrupt line.

All block operations are atomic. This means that the status register state machine status bit always reads 1, ready.

In normal operation, this component has no user-visible registers, but you can read from it as if it is memory.

Programming it or changing the configuration requires a sequence of special write operations, see general flash programming documentation. The component supports Common Flash Interface query operations, which allow drivers to determine the properties of the flash memory.



The model interprets all writes as requests to the programming state machine, and there are many state-machine states that do not support subsequent reads and return 0xdeaddead for them. Therefore, when simulating a ROM, use the `trapwrite=true` option.

---

Use the `diagnostics` parameter to select the level of diagnostic output:

#### Level 0

None.

#### Level 1

Report probable driver error operations:

- Unaligned operations that fault.
- Accesses that the state machine does not expect.
- Transitions of the state machine to unknown states.
- Writes to locked blocks and illegal lock commands.

#### Level 2

Report unimplemented and therefore ignored operations, and log lock commands.

#### Level 3

Warn if a flash write attempts to set bits. The write works if `unphysical_writes=true`.

#### Level 4

Log every read and write.

### Iris and MTI instances for IntelStrataFlashJ3

This model has the following Iris instances:

**Table 3-651: IntelStrataFlashJ3 Iris instances**

InstanceName	ComponentName
IntelStrataFlashJ3	IntelStrataFlashJ3
IntelStrataFlashJ3.map	PVBusMapper
IntelStrataFlashJ3.mbs	PVBusSlave
IntelStrataFlashJ3.rmbs	PVBusSlave

This model has the following MTI trace components:

**Table 3-652: IntelStrataFlashJ3 MTI instances**

InstanceName	ComponentName
IntelStrataFlashJ3.map	PVBusMapper
IntelStrataFlashJ3.mbs	PVBusSlave
IntelStrataFlashJ3.rmbs	PVBusSlave

### Ports for IntelStrataFlashJ3

**Table 3-653: Ports**

Name	Protocol	Type	Description
flashloader	FlashLoaderPort	Slave	-
mem_port	PVDevice	Slave	-
pvbus	PVBus	Slave	-

### Parameters for IntelStrataFlashJ3

#### **diagnostics**

##### Type

int

##### Default value

0x0

##### Description

Diagnostic level.

#### **enable\_read\_status\_logic**

##### Type

bool

##### Default value

0x0

##### Description

Enables logic to handle the status register reads as per the '3 Volt Intel StrataFlash Memory' specification.



**model\_blocklock****Type**

bool

**Default value**

0x0

**Description**

Model per-block locking and set all the blocks to locked state on reset.

**size****Type**

int

**Default value**

0x40000

**Description**

Memory Size.

**trapwrite****Type**

bool

**Default value**

0x0

**Description**

Generate abort on write.

**unphysical\_writes****Type**

bool

**Default value**

0x1

**Description**

Writes to flash are overwrite not AND.

### 3.7.23 Interrupt\_Router

Interrupt Router Registers. This model is written in C++.

#### Ports for Interrupt\_Router

**Table 3-654: Ports**

Name	Protocol	Type	Description
lockdown	Signal	Slave	-
out_interrupts0[64]	Signal	Master	-
out_interrupts1[64]	Signal	Master	-
out_interrupts10[64]	Signal	Master	-
out_interrupts11[64]	Signal	Master	-
out_interrupts12[64]	Signal	Master	-
out_interrupts13[64]	Signal	Master	-
out_interrupts14[64]	Signal	Master	-
out_interrupts15[64]	Signal	Master	-
out_interrupts2[64]	Signal	Master	-
out_interrupts3[64]	Signal	Master	-
out_interrupts4[64]	Signal	Master	-
out_interrupts5[64]	Signal	Master	-
out_interrupts6[64]	Signal	Master	-
out_interrupts7[64]	Signal	Master	-
out_interrupts8[64]	Signal	Master	-
out_interrupts9[64]	Signal	Master	-
pvbus_s	PVBus	Slave	-
reset_signal	Signal	Slave	-
shared_interrupt[428]	Signal	Slave	-
tamper_interrupt	Signal	Master	-

#### Parameters for Interrupt\_Router

##### diagnostics

##### Type

int

##### Default value

0x0

##### Description

Diagnostics to indicate debug level.

**ici\_dst****Type**

string

**Default value**

3,3,3,7,7,7,7,7,7,6,6,0

**Description**

Interrupt Controller Destination The values range from 0 to  $(2^{16} - 1)$ . The bit representation of these values indicate the ICIs possible for each indexed shared interrupt to be routed to.

**ici\_en****Type**

string

**Default value**

1,1,1,1,1,1,1,1,1,1,0

**Description**

Interrupt Controller Enable The values range from 0 to  $(2^{16} - 1)$ . The bit representation of these values indicate the ICIs to which each indexed shared interrupt are routed to.

**lde\_lvl****Type**

int

**Default value**

0x2

**Description**

Lockdown Extension Level 0x0 - LDE.0 is implemented. 0x1 - LDE.1 is implemented. 0x2 - LDE.2 is implemented. All other values are Reserved.

**num\_ici****Type**

int

**Default value**

0x3

**Description**

Number of Interrupt Controllers Interrupt interface.

**num\_shd\_int****Type**

int

**Default value**

0x1

**Description**  
Number of shared interrupts supported.

**ro\_access**

**Type**  
int

**Default value**  
0x0

**Description**  
Stream ID of master.

**rw\_access**

**Type**  
int

**Default value**  
0x0

**Description**  
Stream ID of master with Read Write access.

3.7.24 IoTSS3\_ManagerSecurityController

IoT Subsystem SIE-300 Manager (Master) Security Controller. This model is written in C++.

Iris and MTI instances for IoTSS3\_ManagerSecurityController

This model has the following Iris instances:

Table 3-655: IoTSS3\_ManagerSecurityController Iris instances

InstanceName	ComponentName
IoTSS3_ManagerSecurityController	IoTSS3_ManagerSecurityController
IoTSS3_ManagerSecurityController.pvbusmodifier	PVBusMapper

This model has the following MTI trace components:

Table 3-656: IoTSS3\_ManagerSecurityController MTI instances

InstanceName	ComponentName
IoTSS3_ManagerSecurityController.pvbusmodifier	PVBusMapper

Ports for IoTSS3\_ManagerSecurityController

Table 3-657: Ports

Name	Protocol	Type	Description
cfg_nonsec	ValueState	Slave	-

Name	Protocol	Type	Description
cfg_sec_resp	ValueState	Slave	-
idau_invalidate_region	Value_64	Slave	-
irq	StateSignal	Master	-
pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-

## Parameters for IoTSS3\_ManagerSecurityController

### IRQ\_ENABLE\_RD

#### Type

bool

#### Default value

0x1

#### Description

Interrupt enable read.

### IRQ\_ENABLE\_WR

#### Type

bool

#### Default value

0x1

#### Description

Interrupt enable write.

### diagnostics

#### Type

int

#### Default value

0x0

#### Description

Diagnostics.

## 3.7.25 IoTSS3\_MemoryProtectionController

IoT Subsystem SIE-300 Memory Protection Controller. This model is written in C++.

### Iris and MTI instances for IoTSS3\_MemoryProtectionController

This model has the following Iris instances:

**Table 3-658: IoTSS3\_MemoryProtectionController Iris instances**

InstanceName	ComponentName
IoTSS3_MemoryProtectionController	IoTSS3_MemoryProtectionController
IoTSS3_MemoryProtectionController.bus_mapper	PVBusMapper
IoTSS3_MemoryProtectionController.busslave	PVBusSlave
IoTSS3_MemoryProtectionController.gating_disabled_thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-659: IoTSS3\_MemoryProtectionController MTI instances**

InstanceName	ComponentName
IoTSS3_MemoryProtectionController.bus_mapper	PVBusMapper
IoTSS3_MemoryProtectionController.busslave	PVBusSlave

### Ports for IoTSS3\_MemoryProtectionController

**Table 3-660: Ports**

Name	Protocol	Type	Description
cfg_init_value	ValueState	Slave	-
cfg_sec_resp	ValueState	Slave	-
config_pvbus_s	PVBus	Slave	-
idau_invalidate_region	Value_64	Slave	-
mpc_irq	StateSignal	Master	-
pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-
reset_in	Signal	Slave	-

### Parameters for IoTSS3\_MemoryProtectionController

#### **BLK\_MAX**

##### Type

int

##### Default value

0xffff

##### Description

Maximum block index configuration.

#### **BLK\_SIZE**

##### Type

int

##### Default value

0x3

**Description**

Block size configuration.

**GATE\_PRESENT****Type**

bool

**Default value**

0x1

**Description**

Memory gating logic present/not.

**IRQ\_ENABLE\_RD****Type**

bool

**Default value**

0x1

**Description**

Interrupt enable read.

**IRQ\_ENABLE\_WR****Type**

bool

**Default value**

0x1

**Description**

Interrupt enable write.

**diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

### 3.7.26 IoTSS3\_SecureAccessConfig

IoTSS3 Secure Control Register Block. This model is written in C++.

#### Iris and MTI instances for IoTSS3\_SecureAccessConfig

This model has the following Iris instances:

**Table 3-661: IoTSS3\_SecureAccessConfig Iris instances**

InstanceName	ComponentName
IoTSS3_SecureAccessConfig	IoTSS3_SecureAccessConfig
IoTSS3_SecureAccessConfig.bus_mapper	PVBusMapper
IoTSS3_SecureAccessConfig.busslave_ns	PVBusSlave
IoTSS3_SecureAccessConfig.busslave_s	PVBusSlave
IoTSS3_SecureAccessConfig.idau_busmaster	PVBusMaster
IoTSS3_SecureAccessConfig.p_ahb_bus_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-662: IoTSS3\_SecureAccessConfig MTI instances**

InstanceName	ComponentName
IoTSS3_SecureAccessConfig.bus_mapper	PVBusMapper
IoTSS3_SecureAccessConfig.busslave_ns	PVBusSlave
IoTSS3_SecureAccessConfig.busslave_s	PVBusSlave
IoTSS3_SecureAccessConfig.idau_busmaster	PVBusMaster
IoTSS3_SecureAccessConfig.p_ahb_bus_mapper	PVBusMapper

#### Ports for IoTSS3\_SecureAccessConfig

**Table 3-663: Ports**

Name	Protocol	Type	Description
acc_waitn	ValueState	Master	-
brg_in[32]	StateSignal	Slave	-
brg_out	Signal	Master	-
idau	PVBus	Master	-
mainnspcexp[4]	ValueState	Master	-
mainppcexp[4]	ValueState	Master	-
mem_gating_filter_in	PVBus	Slave	-
mem_gating_filter_out	PVBus	Master	-
mpc_in[32]	StateSignal	Slave	-
mpc_out	Signal	Master	-
msc_in[32]	StateSignal	Slave	-
msc_out	Signal	Master	-
npuspporpl[4]	Signal	Master	-



Name	Protocol	Type	Description
npuspporsl[4]	Signal	Master	-
p_ahb_gating_filter_in	PVBus	Slave	-
p_ahb_gating_filter_out	PVBus	Master	-
periphnsppc0	ValueState	Master	-
periphnsppc1	ValueState	Master	-
periphnsppcexp[4]	ValueState	Master	-
periphpppc0	ValueState	Master	-
periphpppc1	ValueState	Master	-
periphpppcexp[4]	ValueState	Master	-
ppc_in[32]	StateSignal	Slave	-
ppc_out	Signal	Master	-
pvbus_nonsecure	PVBus	Slave	-
pvbus_secure	PVBus	Slave	-
reset_in	Signal	Slave	-
security_resp	ValueState	Master	-

## Parameters for IoTSS3\_SecureAccessConfig

### **CODENSC**

#### Type

bool

#### Default value

0x0

#### Description

Whether 0x10000000..0x1FFFFFFF is non-secure-callable.

### **DISABLE\_GATING**

#### Type

bool

#### Default value

0x0

#### Description

Disable Memory gating logic.

### **IGNORE\_MEM\_MAP**

#### Type

bool

#### Default value

0x0

**Description**

Ignore Memory mapping logic.

**MAINPPCEXP\_DIS0****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the MAINNSPPCEXP0 and MAINPPPCEXP0 buses.

**MAINPPCEXP\_DIS1****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the MAINNSPPCEXP1 and MAINPPPCEXP1 buses.

**MAINPPCEXP\_DIS2****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the MAINNSPPCEXP2 and MAINPPPCEXP2 buses.

**MAINPPCEXP\_DIS3****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the MAINNSPPCEXP3 and MAINPPPCEXP3 buses.

**PERIPHPPCEXP\_DIS0****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the PERIPHNSPPCEXP0 and PERIPHPPPCEXP0 buses.

**PERIPHPPPCEXP\_DIS1****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the PERIPHNSPPCEXP1 and PERIPHPPPCEXP1 buses.

**PERIPHPPPCEXP\_DIS2****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the PERIPHNSPPCEXP2 and PERIPHPPPCEXP2 buses.

**PERIPHPPPCEXP\_DIS3****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the PERIPHNSPPCEXP3 and PERIPHPPPCEXP3 buses.

**RAMNSC****Type**

bool

**Default value**

0x0

**Description**

Whether 0x30000000..0x3FFFFFFF is non-secure-callable.

**diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

### 3.7.27 Juno\_sysregs

IOFPGA system register unit. This model is written in C++.

**Iris and MTI instances for Juno\_sysregs**

This model has the following Iris instances:

**Table 3-664: Juno\_sysregs Iris instances**

InstanceName	ComponentName
Juno_sysregs	Juno_sysregs
Juno_sysregs.pvbuslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-665: Juno\_sysregs MTI instances**

InstanceName	ComponentName
Juno_sysregs.pvbuslave	PVBusSlave

**Ports for Juno\_sysregs****Table 3-666: Ports**

Name	Protocol	Type	Description
clock_100Hz	ClockSignal	Slave	-
clock_24Mhz	ClockSignal	Slave	-
ethernet_irq	Signal	Master	-
mmc_presence	StateSignal	Slave	-
pb_irq	Signal	Master	-
pvbus	PVBus	Slave	-
rtcc_irq	Signal	Master	-
tile1_irq	Signal	Master	-
usb_irq	Signal	Master	-

Parameters for Juno\_sysregs

**diagnostics**

**Type**  
int

**Default value**  
0x0

**Description**  
Diagnostics.

**pcie\_mac**

**Type**  
int

**Default value**  
0x2f7000001

**Description**  
PCIe MAC address.

**rev**

**Type**  
int

**Default value**  
0x0

**Description**  
Board revision.

3.7.28 Kits2\_Timer

Kits2 Timer. This model is written in C++.

Iris and MTI instances for Kits2\_Timer

This model has the following Iris instances:

Table 3-667: Kits2\_Timer Iris instances

InstanceName	ComponentName
Kits2_Timer	Kits2_Timer
Kits2_Timer.clk_div	ClockDivider
Kits2_Timer.counter	CounterModule

This model has the following MTI trace components:

**Table 3-668: Kits2\_Timer MTI instances**

InstanceName	ComponentName
Kits2_Timer.clk_div	ClockDivider

**Ports for Kits2\_Timer****Table 3-669: Ports**

Name	Protocol	Type	Description
clock	ClockSignal	Slave	-
enable	Signal	Slave	-
irq_out	Signal	Master	-
timer_freq	ClockRateControl	Slave	-
timer_value	ValueState	Slave	-

**Parameters for Kits2\_Timer****clk\_div.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clk\_div.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**counter.diagnostics****Type**

int

**Default value**

0x2

**Description**

Diagnostics.

**diagnostics**

**Type**  
int

**Default value**  
0x0

**Description**  
Diagnostics.

**timer\_interval**

**Type**  
int

**Default value**  
0x0

**Description**  
Timer ticks to count before firing interrupt.

3.7.29 LS64TestingFIFO

FIFO peripheral supporting LS64 accesses for testing purposes. This model is written in LISA+.

About LD64TestingFIFO

LS64TestingFIFO is a LISA component for testing the FEAT\_LS64 architectural feature. It accepts ST64B instructions and places the supplied data into a configurable buffer. LD64B instructions can then read this data out of the FIFO. The value returned can configurably be bitwise inverted.

It also supports the ST64BV variants where success and failure are reported by a return result rather than by transaction success and failure.

Iris and MTI instances for LS64TestingFIFO

This model has the following Iris instances:

Table 3-670: LS64TestingFIFO Iris instances

InstanceName	ComponentName
LS64TestingFIFO	LS64TestingFIFO
LS64TestingFIFO.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Table 3-671: LS64TestingFIFO MTI instances

InstanceName	ComponentName
LS64TestingFIFO.pvbusslave	PVBusSlave

## Ports for LS64TestingFIFO

**Table 3-672: Ports**

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	Bus subordinate interface.

## Parameters for LS64TestingFIFO

### buffer\_size

#### Type

int

#### Default value

0x2

#### Description

The number of 64-byte slots in the FIFO.

### op\_type

#### Type

int

#### Default value

0x1

#### Description

The operation performed on the 64-byte transaction data 0 - None, 1 - Bitwise Negate.

## 3.7.30 LabellerIdauSecurity

This model is written in C++.

### Iris and MTI instances for LabellerIdauSecurity

This model has the following Iris instances:

**Table 3-673: LabellerIdauSecurity Iris instances**

InstanceName	ComponentName
LabellerIdauSecurity	LabellerIdauSecurity
LabellerIdauSecurity.idau_busmaster	PVBusMaster
LabellerIdauSecurity.pvbusmodifier	PVBusMapper
LabellerIdauSecurity.remap_busmaster	PVBusMaster

This model has the following MTI trace components:



**Table 3-674: LabellerIdauSecurity MTI instances**

InstanceName	ComponentName
LabellerIdauSecurity.idau_busmaster	PVBusMaster
LabellerIdauSecurity.pvbusmodifier	PVBusMapper
LabellerIdauSecurity.remap_busmaster	PVBusMaster

### Ports for LabellerIdauSecurity

**Table 3-675: Ports**

Name	Protocol	Type	Description
idau	PVBus	Master	-
idau_invalidate_region	Value_64	Slave	-
pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-

### Parameters for LabellerIdauSecurity

#### diagnostics

##### Type

int

##### Default value

0x0

##### Description

Diagnostics.

## 3.7.31 MemoryMappedGenericTimer

ARM Generic Timer. This model is written in C++.

### Iris and MTI instances for MemoryMappedGenericTimer

This model has the following Iris instances:

**Table 3-676: MemoryMappedGenericTimer Iris instances**

InstanceName	ComponentName
MemoryMappedGenericTimer	MemoryMappedGenericTimer
MemoryMappedGenericTimer.busbase0	PVBusSlave
MemoryMappedGenericTimer.busbase1	PVBusSlave
MemoryMappedGenericTimer.busbase10	PVBusSlave
MemoryMappedGenericTimer.busbase11	PVBusSlave
MemoryMappedGenericTimer.busbase12	PVBusSlave
MemoryMappedGenericTimer.busbase13	PVBusSlave
MemoryMappedGenericTimer.busbase14	PVBusSlave

InstanceName	ComponentName
MemoryMappedGenericTimer.busbase15	PVBusSlave
MemoryMappedGenericTimer.busbase2	PVBusSlave
MemoryMappedGenericTimer.busbase3	PVBusSlave
MemoryMappedGenericTimer.busbase4	PVBusSlave
MemoryMappedGenericTimer.busbase5	PVBusSlave
MemoryMappedGenericTimer.busbase6	PVBusSlave
MemoryMappedGenericTimer.busbase7	PVBusSlave
MemoryMappedGenericTimer.busbase8	PVBusSlave
MemoryMappedGenericTimer.busbase9	PVBusSlave
MemoryMappedGenericTimer.busctlbase	PVBusSlave

This model has the following MTI trace components:

**Table 3-677: MemoryMappedGenericTimer MTI instances**

InstanceName	ComponentName
MemoryMappedGenericTimer.busbase0	PVBusSlave
MemoryMappedGenericTimer.busbase1	PVBusSlave
MemoryMappedGenericTimer.busbase10	PVBusSlave
MemoryMappedGenericTimer.busbase11	PVBusSlave
MemoryMappedGenericTimer.busbase12	PVBusSlave
MemoryMappedGenericTimer.busbase13	PVBusSlave
MemoryMappedGenericTimer.busbase14	PVBusSlave
MemoryMappedGenericTimer.busbase15	PVBusSlave
MemoryMappedGenericTimer.busbase2	PVBusSlave
MemoryMappedGenericTimer.busbase3	PVBusSlave
MemoryMappedGenericTimer.busbase4	PVBusSlave
MemoryMappedGenericTimer.busbase5	PVBusSlave
MemoryMappedGenericTimer.busbase6	PVBusSlave
MemoryMappedGenericTimer.busbase7	PVBusSlave
MemoryMappedGenericTimer.busbase8	PVBusSlave
MemoryMappedGenericTimer.busbase9	PVBusSlave
MemoryMappedGenericTimer.busctlbase	PVBusSlave

## Ports for MemoryMappedGenericTimer

**Table 3-678: Ports**

Name	Protocol	Type	Description
cntpsirq[8]	Signal	Master	-
cntvalueb	CounterInterface	Slave	-
pvbus_base_s[8]	PVBus	Slave	-
pvbus_ctlbase_s	PVBus	Slave	-

Name	Protocol	Type	Description
pvbus_el0base_s[8]	PVBus	Slave	-
timer_reset	Signal	Slave	-

## Parameters for MemoryMappedGenericTimer

### bypass\_ctlbase

#### Type

bool

#### Default value

0x0

#### Description

Bypass CNTBase Access Control. Enable if only timer frame feature is required without CNTBase access control.

### cntel0acr\_implemented

#### Type

int

#### Default value

0x0

#### Description

A bit-field of 8 bits, where bit {n} enables CNTELOACR for timer frame {n}.

### diagnostics

#### Type

int

#### Default value

0x0

#### Description

Diagnostics.

### frame\_security

#### Type

string

#### Default value

""

#### Description

Hard-wired/configurable security for frames (N/S/X, one character per timer frame).

**num\_timers****Type**

int

**Default value**

0x1

**Description**

Number of timer frames.

### 3.7.32 MemoryMappedGenericWatchdog

ARM Generic Watchdog. This model is written in C++.

#### About MemoryMappedGenericWatchdog

This is a high-level watchdog that generates two interrupts rather than an interrupt followed by a reset.

#### Iris and MTI instances for MemoryMappedGenericWatchdog

This model has the following Iris instances:

**Table 3-679: MemoryMappedGenericWatchdog Iris instances**

InstanceName	ComponentName
MemoryMappedGenericWatchdog	MemoryMappedGenericWatchdog
MemoryMappedGenericWatchdog.busctlbase	PVBusSlave
MemoryMappedGenericWatchdog.busrefbase	PVBusSlave

This model has the following MTI trace components:

**Table 3-680: MemoryMappedGenericWatchdog MTI instances**

InstanceName	ComponentName
MemoryMappedGenericWatchdog.busctlbase	PVBusSlave
MemoryMappedGenericWatchdog.busrefbase	PVBusSlave

#### Ports for MemoryMappedGenericWatchdog

**Table 3-681: Ports**

Name	Protocol	Type	Description
cntvalueb	CounterInterface	Slave	-
ctl_pvbus_s	PVBus	Slave	-
ref_pvbus_s	PVBus	Slave	-
reset_in	Signal	Slave	-
WS0	Signal	Master	-
WS1	Signal	Master	-

## Parameters for MemoryMappedGenericWatchdog

### **NONSECURE**

**Type**

bool

**Default value**

0x0

**Description**

Non-Secure.

### **arch\_version**

**Type**

int

**Default value**

0x0

**Description**

Architecture version. Available 0 and 1.

### **diagnostics**

**Type**

int

**Default value**

0x0

**Description**

Diagnostics.

### **product\_id**

**Type**

int

**Default value**

0x0

**Description**

Product Identifier.

## 3.7.33 NonVolatileCounter

Trusted Non-Volatile Counter unit. This model is written in C++.

### **Iris and MTI instances for NonVolatileCounter**

This model has the following Iris instances:

**Table 3-682: NonVolatileCounter Iris instances**

InstanceName	ComponentName
NonVolatileCounter	NonVolatileCounter
NonVolatileCounter.pvbusslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-683: NonVolatileCounter MTI instances**

InstanceName	ComponentName
NonVolatileCounter.pvbusslave	PVBusSlave

### Ports for NonVolatileCounter

**Table 3-684: Ports**

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	-

### Parameters for NonVolatileCounter

#### **diagnostics**

##### Type

int

##### Default value

0x0

##### Description

Diagnostics.

#### **rst\_non\_tz\_fw\_cnt**

##### Type

int

##### Default value

0x0

##### Description

Value of NON\_TZ\_FW\_CNT at reset.

#### **rst\_tz\_fw\_cnt**

##### Type

int

##### Default value

0x0

##### Description

Value of TZ\_FW\_CNT at reset.

**secure**

**Type**

int

**Default value**

0x1

**Description**

Instantiate model as Secure (1) or NS (0).

**version**

**Type**

string

**Default value**

"r0"

**Description**

Version of the model functionality. Valid values are r0 and r1.

3.7.34 PCIeATC

This model is written in C++.

About PCIeATC

This component is for validation only. It is not directly suitable for use as an ATC. It is used for testing the ATC implementation of `pcie_atc_if` produced by `make_PCIeATC_v0()`.

Iris and MTI instances for PCIeATC

This model has the following Iris instances:

Table 3-685: PCIeATC Iris instances

InstanceName	ComponentName
PCIeATC	validation_atc
PCIeATC.ExportTest.PCIeATC.mapper	PVBusMapper
PCIeATC.pvbus_id_routed_s[0]	PVBusSlave

This model has the following MTI trace components:

Table 3-686: PCIeATC MTI instances

InstanceName	ComponentName
PCIeATC	atc
PCIeATC.ExportTest.PCIeATC.mapper	PVBusMapper
PCIeATC.pvbus_id_routed_s[0]	PVBusSlave

## Ports for PCIeATC

**Table 3-687: Ports**

Name	Protocol	Type	Description
atc	PCIeATC_get_if	Slave	-
disable_PRI_and_set_RF	Signal	Master	This is pulsed (set, then clear) when a condition occurs that causes a Response Failure. The correct response of the PCIe device is to disable PRI and to set the RF bit in the PRI header.
identify	SMMUv3AEMIdentifyProtocol	Master	The user has a chance to determine how the substreamid is extracted from the transactions received on pvbus_s by using this port. If it is unimplemented then the ATC will use the default policy identified in SMMUv3_FOR_PCIE.lisa <201601041554/>
pvbus_id_routed_s	PVBus	Slave	-
pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-
uprgi	Signal	Master	This is pulsed (set, then clear) when an Unrecognised PRG Index is received. In a real PCIe device this would set the UPRGI bit in the PRI header.

## Parameters for PCIeATC

### atc\_size

#### Type

int

#### Default value

0x0

#### Description

The maximum number of ATC entries. 0 is effectively a large number.

### seed

#### Type

int

#### Default value

0x12345678

#### Description

Seed for a random number generator.

## 3.7.35 PLLClockControl

Clock Rate Control. This model is written in C++.

### Iris and MTI instances for PLLClockControl

This model has the following Iris instances:



**Table 3-688: PLLClockControl Iris instances**

InstanceName	ComponentName
PLLClockControl	PLLClockControl
PLLClockControl.clock_ctl	ClockDivider
PLLClockControl.pllclk_div	ClockDivider

This model has the following MTI trace components:

**Table 3-689: PLLClockControl MTI instances**

InstanceName	ComponentName
PLLClockControl.clock_ctl	ClockDivider
PLLClockControl.pllclk_div	ClockDivider

## Ports for PLLClockControl

**Table 3-690: Ports**

Name	Protocol	Type	Description
clk_en	Signal	Slave	-
clk_in	ClockSignal	Slave	-
clk_out	ClockSignal	Master	-
clk_rate	ClockRateControl	Slave	-
clk_sel	Value	Slave	-
dvfs_freq_in	ValueState	Slave	-
lock	Signal	Master	-
refclk_in	ClockSignal	Slave	-
unlock	Signal	Master	-

## Parameters for PLLClockControl

### diagnostics

#### Type

int

#### Default value

0x0

#### Description

Diagnostics.

### 3.7.36 PPUMultiThreadModeSwitch

PPU mode switch between single-thread mode and multi-thread mode. Support up to 8 cores and thread number per core is no more than 2. This model is written in C++.

#### Ports for PPUMultiThreadModeSwitch

**Table 3-691: Ports**

Name	Protocol	Type	Description
pchannel_from_ppu_s[8]	PChannel	Slave	-
pchannel_to_cpu_m[8]	PChannel	Master	-
wakerequest_from_gic_s[16]	Signal	Slave	-
wakerequest_to_ppu_m[8]	Signal	Master	-

#### Parameters for PPUMultiThreadModeSwitch

##### mt\_mode

##### Type

bool

##### Default value

0x0

##### Description

Multi-threaded mode.

### 3.7.37 PS2Keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050\_KMI component. This model is written in LISA+.

#### Iris and MTI instances for PS2Keyboard

This model has the following Iris instances:

**Table 3-692: PS2Keyboard Iris instances**

InstanceName	ComponentName
PS2Keyboard	PS2Keyboard
PS2Keyboard.ps2_clocktimer	ClockTimerThread
PS2Keyboard.ps2_clocktimer.timer	ClockTimerThread64
PS2Keyboard.ps2_clocktimer.timer.thread	SchedulerThread
PS2Keyboard.ps2_clocktimer.timer.thread_event	SchedulerThreadEvent

## Ports for PS2Keyboard

**Table 3-693: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	Clock signal to rate at which PS2Data signals are generated.
keyboard	<a href="#">KeyboardStatus</a>	Slave	Receives keyboard input from, for example, the Visualisation component.
ps2	<a href="#">PS2Data</a>	Master	Connection to the PS/2 controller, for example, the PL050_KMI.

### 3.7.38 PS2Mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050\_KMI component. This model is written in C++.

#### Iris and MTI instances for PS2Mouse

This model has the following Iris instances:

**Table 3-694: PS2Mouse Iris instances**

InstanceName	ComponentName
PS2Mouse	PS2Mouse

## Ports for PS2Mouse

**Table 3-695: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	-
mouse	<a href="#">MouseStatus</a>	Slave	-
ps2	<a href="#">PS2Data</a>	Master	-

### 3.7.39 PVBUSGICv3Comms

GICv3 Component for conversion between GICv3Comms protocol and PVBUS. This model is written in C++.

#### Iris and MTI instances for PVBUSGICv3Comms

This model has the following Iris instances:

**Table 3-696: PVBUSGICv3Comms Iris instances**

InstanceName	ComponentName
PVBUSGICv3Comms	PVBUSGICv3Comms
PVBUSGICv3Comms.bus_slave	PVBUSSlave

This model has the following MTI trace components:

**Table 3-697: PVBUSGICv3Comms MTI instances**

InstanceName	ComponentName
PVBUSGICv3Comms	PVBUSGICv3Comms
PVBUSGICv3Comms.bus_slave	PVBUSSlave

### Ports for PVBUSGICv3Comms

**Table 3-698: Ports**

Name	Protocol	Type	Description
axi_master_id_m[256]	Value	Master	-
distributor_m[256]	GICv3Comms	Master	-
pvbuss_m	PVBUS	Master	-
pvbuss_s	PVBUS	Slave	-

## 3.7.40 PVMetaDataController

A simulation-only (not in hardware) component that can service metadata requests for transactions on PVBUS. This model is written in C++.

### About PVMetaDataController

This component represents an entity in a memory system that responds to requests for manipulating metadata during a bus transaction and/or as part of Armv8.5-A instruction execution.

This component is intended to be an `sg::Component` which can be instantiated and connected in a platform.

MetaDataController is a `pv::RemapTransactionIntermediary` as it needs to intercept bus transactions to apply metadata operations and set up DMI to metadata memory, that is, assign `MetaDataPayload_t.set_dmi()`.

By being a `pv::RemapTransactionIntermediary` and using DMIs for data and metadata, it has a very small impact on simulation speed.



Note

The Armv8.5-A specification mentions that Colour-Check, which is a certain kind of metadata operation, happens in the physical memory system and a Processing Element (PE) only cares about the result of such an operation.

However, doing this for all memory accesses drastically slows down the simulation. While the first memory access of a `MetaDataPage_t` comes to `MetaDataController`, the rest of the accesses for addresses in this page are made through `MetaDataDMI_t`. This essentially means that Colour-Check is done in this class only once for every `MetaDataPage_t`. On other occasions, the intention is that the checks are done by the holder of `MetaDataDMI_t`.

## Iris and MTI instances for PVMetaDataController

This model has the following Iris instances:

**Table 3-699: PVMetaDataController Iris instances**

InstanceName	ComponentName
PVMetaDataController	MetaDataController

## Ports for PVMetaDataController

**Table 3-700: Ports**

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-

## Parameters for PVMetaDataController

### init\_value

#### Type

int

#### Default value

0xd

#### Description

Initialize metadata memory with this value. If one of init\_values\_json or init\_values\_json\_file is specified, this value applies only to any metadata not specified in the JSON.

### init\_values\_json

#### Type

string

#### Default value

""

A JSON value describing initial metadata values. Mutually exclusive with init\_values\_json\_file .

The format is as follows:

```
{ "regions": [{ "begin": 0x0,
                "end": 0x10000,
                "mte_tag": 0xa},
               { "begin": 0x20000,
                "end": 0x50000,
                "mte_tag": 0xc}]
}
```

**init\_values\_json\_file****Type**

string

**Default value**

""

Path to a JSON file with initial metadata values. Mutually exclusive with `init_values_json`. The format is as follows:

```
{ "regions": [{ "begin": 0x0,
                "end": 0x10000,
                "mte_tag": 0xa},
               { "begin": 0x20000,
                "end": 0x50000,
                "mte_tag": 0xc}]
}
```

**is\_enabled****Type**

bool

**Default value**

0x0

**Description**

If false, disables the MetaData controller functionality, and makes the component invisible to passing transactions.

**mte\_tag\_carveout\_json****Type**

string

**Default value**

""

JSON string that specifies the PA range of the tag carveout regions and the beginning of the PA range for which they provide tag storage.

If `pa_regions_with_metadata_storage` defines which regions can have metadata, the tag carveout regions cannot overlap them, and each tagged region must be entirely covered by one of them.

The block size must be  $\geq 64$  bytes and a power of 2, defaulting to 4KiB. The maximum block size supported is 4KiB.

The carveout region size must be  $\geq 4\text{KiB}$  and a power of 2, and determines the size of the corresponding tagged region.

```
{ "regions": [{ "begin": 0x0,      "tag_carveout_region":
[0xffffffff00000, 0xffffffff00fff]},
    { "begin": 0x20000, "tag_carveout_region":
[0xffffffff01000, 0xffffffff01fff], "block_size": 0x100},
    { "begin": 0x100000, "tag_carveout_region":
[0xffffffff08000, 0xffffffff0Bfff], "block_size": 0x2000}]}
```

### **mte\_tag\_carveout\_json\_file**

#### **Type**

string

#### **Default value**

""

Path to a file which contains the JSON string that specifies the PA range of the tag carveout regions and the beginning of the PA range for which they provide tag storage.

If `pa_regions_with_metadata_storage` defines which regions can have metadata, the tag carveout regions cannot overlap them, and each tagged region must be entirely covered by one of them.

The block size must be  $\geq 64$  bytes and a power of 2, defaulting to  $4\text{KiB}$ . The maximum block size supported is  $4\text{KiB}$ .

The carveout region size must be  $\geq 4\text{KiB}$  and a power of 2, and determines the size of the corresponding tagged region.

```
{ "regions": [{ "begin": 0x0,      "tag_carveout_region":
[0xffffffff00000, 0xffffffff00fff]},
    { "begin": 0x20000, "tag_carveout_region":
[0xffffffff01000, 0xffffffff01fff], "block_size": 0x100},
    { "begin": 0x100000, "tag_carveout_region":
[0xffffffff08000, 0xffffffff0Bfff], "block_size": 0x2000}]}
```

Only one of `mte_tag_carveout_json` and `mte_tag_carveout_json_file` can be used.

### **mte\_tag\_carveout\_tag\_order**

#### **Type**

string

#### **Default value**

"little-endian"

**Description**

Order of the tags within the MTE tag carveout blocks. This can be little-endian (same order as the corresponding tagged data) or big-endian (reverse order). The parameter accepts both '-' and '\_', so 'little-endian', 'big-endian', 'little\_endian' and 'big\_endian' are all valid. .

**pa\_regions\_with\_metadata\_storage**

**Type**

string

**Default value**

""

Specify the address region where the metadata storage is available for each PAS in a JSON format.

If the PAS does not have a region specified, the PAS has metadata storage for all of the space.

The regions are defined by begin and end\_incl addresses. Example:

```
{ "ns": [0xa0000000, 0xa0000fff],  
  "s" : [0xb0000000, 0xb0000fff],  
  "rl": [0xc0000000, 0xc0000fff],  
  "rt": [0xd0000000, 0xd0000fff]}
```

ns: non-secure, s: secure, rl: realm, rt: root

**3.7.41 PchannelListener**

Provides a dummy PChannel device to accept all request. This model is written in C++.

**Ports for PchannelListener**

Table 3-701: Ports

Name	Protocol	Type	Description
dev_pchannel_s	PChannel	Slave	-

**Parameters for PchannelListener**

**diagnostics**

**Type**

int

**Default value**

0x0

**Description**

Diagnostics.



### 3.7.42 PowerStateGate

Power State Gate to filter the access to SYSTOP domain. This model is written in C++.

#### Iris and MTI instances for PowerStateGate

This model has the following Iris instances:

**Table 3-702: PowerStateGate Iris instances**

InstanceName	ComponentName
PowerStateGate	PowerStateGate
PowerStateGate.filter	PVBusMapper

This model has the following MTI trace components:

**Table 3-703: PowerStateGate MTI instances**

InstanceName	ComponentName
PowerStateGate.filter	PVBusMapper

#### Ports for PowerStateGate

**Table 3-704: Ports**

Name	Protocol	Type	Description
powerdown	Signal	Slave	-
pdbus_m	PVBus	Master	-
pdbus_s	PVBus	Slave	-

#### Parameters for PowerStateGate

##### diagnostics

###### Type

int

###### Default value

0x0

###### Description

Diagnostics.

##### gate\_behaviour

###### Type

int

###### Default value

0x0

**Description**  
Gate behaviour when power is down, 0=abort, 1=ignore.

3.7.43 RAMDevice

RAM device, can be dynamic or static ram. This model is written in LISA+.

**About RAMDevice**  
As a generic device, this component does not have a hardware revision code.

**Iris and MTI instances for RAMDevice**  
This model has the following Iris instances:

Table 3-705: RAMDevice Iris instances

InstanceName	ComponentName
RAMDevice	RAMDevice
RAMDevice.bus_slave	PVBusSlave

This model has the following MTI trace components:

Table 3-706: RAMDevice MTI instances

InstanceName	ComponentName
RAMDevice.bus_slave	<a href="#">PVBusSlave</a>

Ports for RAMDevice

Table 3-707: Ports

Name	Protocol	Type	Description
pvbus	<a href="#">PVBus</a>	Slave	Bus slave interface.

Parameters for RAMDevice

**enable\_atomic\_ops**  
Type  
bool  
Default value  
0x0  
Description  
Supports Atomic Operations.

**fill11**  
Type  
int

**Default value**

0x0

**Description**

Fill pattern 1, initialise memory at start of simulation with alternating fill1, fill2 pattern.

**fill2****Type**

int

**Default value**

0x0

**Description**

Fill pattern 2, initialise memory at start of simulation with alternating fill1, fill2 pattern.

**read\_latency****Type**

int

**Default value**

0x0

**Description**

Memory read latency (ps/byte).

**size****Type**

int

**Default value**

0x100000000

**Description**

Memory Size.

**write\_latency****Type**

int

**Default value**

0x0

**Description**

Memory write latency (ps/byte).

## 3.7.44 ROM

Simple ROM device. This model is written in LISA+.

### Iris and MTI instances for ROM

This model has the following Iris instances:

**Table 3-708: ROM Iris instances**

InstanceName	ComponentName
ROM	ROM
ROM.bus_mapper	PVBusMapper
ROM.bus_slave	PVBusSlave

This model has the following MTI trace components:

**Table 3-709: ROM MTI instances**

InstanceName	ComponentName
ROM.bus_mapper	PVBusMapper
ROM.bus_slave	PVBusSlave

### Ports for ROM

**Table 3-710: Ports**

Name	Protocol	Type	Description
parity_error_out	Signal	Master	-
pvbus	PVBus	Slave	-

### Parameters for ROM

#### **abort\_writes**

##### Type

bool

##### Default value

0x0

##### Description

Abort writes instead of ignoring them.

#### **log2\_size**

##### Type

int

##### Default value

0x14

**Description**

Log2 size (bytes) e.g. 20 is 1 MiB.

**parity\_enabled**

**Type**

bool

**Default value**

0x0

**Description**

Parity Check Enabled on ROM data: If this parameter is enabled, Model assumes that ROM binary will have data + parity.

**raw\_image**

**Type**

string

**Default value**

""

**Description**

Raw image file to load at init time.

3.7.45 RSE\_Integ\_Regs

RSE Integration Layer Registers. This model is written in C++.

Iris and MTI instances for RSE\_Integ\_Regs

This model has the following Iris instances:

Table 3-711: RSE\_Integ\_Regs Iris instances

InstanceName	ComponentName
RSE_Integ_Regs	RSE_Integration_Registers
RSE_Integ_Regs.ClockDivider	ClockDivider
RSE_Integ_Regs.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Table 3-712: RSE\_Integ\_Regs MTI instances

InstanceName	ComponentName
RSE_Integ_Regs.ClockDivider	ClockDivider
RSE_Integ_Regs.PVBusSlave	PVBusSlave

## Ports for RSE\_Integ\_Regs

Table 3-713: Ports

Name	Protocol	Type	Description
EXTMCPRESETn	Signal	Master	-
EXTSCPRESETn	Signal	Master	-
MCP_ATU_AP	Signal	Master	-
MCP_RAS_ERR_CLEAR	Signal	Master	-
pvbus_s	PVBus	Slave	-
REFCLK	ClockSignal	Slave	-
reset_in	Signal	Slave	-
RSECORECLK	ClockSignal	Master	-
SCP_ATU_AP	Signal	Master	-
SCP_RAS_ERR_CLEAR	Signal	Master	-
SYSPLLCLK	ClockSignal	Slave	-

## Parameters for RSE\_Integ\_Regs

**chip\_id****Type**

int

**Default value**

0x0

**Description**

RSE Integration Register Chip ID.

**diagnostics****Type**

int

**Default value**

0x0

**Description**

RSE Integration Registers Diagnostics.

**multichip\_mode****Type**

bool

**Default value**

0x0

**Description**

RSE Integration Register multichip mode.

### 3.7.46 RandomNumberGenerator

Random Number Generator unit. This model is written in C++.

#### Iris and MTI instances for RandomNumberGenerator

This model has the following Iris instances:

**Table 3-714: RandomNumberGenerator Iris instances**

InstanceName	ComponentName
RandomNumberGenerator	RandomNumberGenerator
RandomNumberGenerator.pvbuslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-715: RandomNumberGenerator MTI instances**

InstanceName	ComponentName
RandomNumberGenerator.pvbuslave	PVBusSlave

#### Ports for RandomNumberGenerator

**Table 3-716: Ports**

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	Bus slave interface.
RNG_intr	Signal	Master	Interrupt output.

#### Parameters for RandomNumberGenerator

##### diagnostics

###### Type

int

###### Default value

0x0

###### Description

Diagnostics.

##### seed

###### Type

int

###### Default value

0x0

###### Description

Random number seed.

### 3.7.47 RealTimeLimiter

Real Time Limiter. This model is written in LISA+.

#### Iris and MTI instances for RealTimeLimiter

This model has the following Iris instances:

Table 3-717: RealTimeLimiter Iris instances

InstanceName	ComponentName
RealTimeLimiter	RealTimeLimiter
RealTimeLimiter.divider	ClockDivider

This model has the following MTI trace components:

Table 3-718: RealTimeLimiter MTI instances

InstanceName	ComponentName
RealTimeLimiter.divider	ClockDivider

#### Ports for RealTimeLimiter

Table 3-719: Ports

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock input.

#### Parameters for RealTimeLimiter

##### ENABLE

Type

bool

Default value

0x0

Description

Rate limit simulation.

##### RELATIVE\_SPEED

Type

int

Default value

0x64

Description

Rate limit to at most this percentage of real time (100: limit to wall clock rate).



**divider.div**

**Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**3.7.48 RealtimeClockTimer**

Host Time Based Timer Module for Generic Timers. This model is written in C++.

**Ports for RealtimeClockTimer**

Table 3-720: Ports

Name	Protocol	Type	Description
set_frequency	Value_64	Slave	-
timer_callback	TimerCallback64	Master	-
timer_control	TimerControl64	Slave	-

**3.7.49 RemapDecoder**

The component that provides support for dynamically remappable regions of memory. This model is written in LISA+.

**Iris and MTI instances for RemapDecoder**

This model has the following Iris instances:

Table 3-721: RemapDecoder Iris instances

InstanceName	ComponentName
RemapDecoder	RemapDecoder
RemapDecoder.bus_switch	TZSwitch
RemapDecoder.bus_switch.pvbus_mapper	PVBusMapper

This model has the following MTI trace components:

Table 3-722: RemapDecoder MTI instances

InstanceName	ComponentName
RemapDecoder.bus_switch.pvbus_mapper	PVBusMapper

## Ports for RemapDecoder

**Table 3-723: Ports**

Name	Protocol	Type	Description
control	TZSwitchControl	Broadcast	-
input	PVBus	Slave	Incoming bus transactions (connected straight to TZSwitch).
output_remap_clear	PVBus	Master	Outgoing bus transactions when remap is clear.
output_remap_set	PVBus	Master	Outgoing bus transactions when remap is set.
remap	StateSignal	Slave	Remapping control.

## Parameters for RemapDecoder

### **bus\_switch.normal**

#### Type

int

#### Default value

0x2

#### Description

Normal Port.

### **bus\_switch.secure**

#### Type

int

#### Default value

0x1

#### Description

Secure Port.

## 3.7.50 RootKeyStorage

Trusted Root-Key Storage unit. This model is written in C++.

### Iris and MTI instances for RootKeyStorage

This model has the following Iris instances:

**Table 3-724: RootKeyStorage Iris instances**

InstanceName	ComponentName
RootKeyStorage	RootKeyStorage
RootKeyStorage.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Table 3-725: RootKeyStorage MTI instances

InstanceName	ComponentName
RootKeyStorage.pvbuslave	PVBusSlave

Ports for RootKeyStorage

Table 3-726: Ports

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	-

Parameters for RootKeyStorage

diagnostics

Type

int

Default value

0x0

Description

Diagnostics.

hw\_unique\_key

Type

string

Default value

"00000000 00000000 00000000 00000000"

Description

Hardware Unique Key (128-bit, 4 std::hex words).

hw\_unique\_key\_hex

Type

string

Default value

""

Description

Hardware Unique Key (128-bit, little-endian std::hex byte stream).

private\_key

Type

string

Default value

"00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000"

**Description**

Private Endorsement Key (256-bit, 8 std::hex words).

**private\_key\_hex****Type**

string

**Default value**

""

**Description**

Private Key (256-bit, little-endian std::hex byte stream).

**public\_key****Type**

string

**Default value**

"00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000"

**Description**

Public Key (256-bit, 8 std::hex words).

**public\_key\_hex****Type**

string

**Default value**

""

**Description**

Public Key (256-bit, little-endian std::hex byte stream).

**ss\_key****Type**

string

**Default value**

"00000000 00000000 00000000 00000000"

**Description**

Secret Symmetric Key (128-bit, 4 std::hex words).

**ss\_key\_hex****Type**

string

**Default value**

""

**Description**  
Secret Symmetric Key (128-bit, little-endian std::hex byte stream).

**version**

**Type**  
string

**Default value**  
"r1"

**Description**  
Version of the model functionality. Valid values are r0 and r1.

3.7.51 SI\_System\_Ctrl\_Regs

Safety Island System Control Registers. This model is written in C++.

Iris and MTI instances for SI\_System\_Ctrl\_Regs

This model has the following Iris instances:

Table 3-727: SI\_System\_Ctrl\_Regs Iris instances

InstanceName	ComponentName
SI_System_Ctrl_Regs	SI_System_Control_Registers
SI_System_Ctrl_Regs.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Table 3-728: SI\_System\_Ctrl\_Regs MTI instances

InstanceName	ComponentName
SI_System_Ctrl_Regs.pvbusslave	PVBusSlave

Ports for SI\_System\_Ctrl\_Regs

Table 3-729: Ports

Name	Protocol	Type	Description
cpuhalt_m[7]	Signal	Master	-
pvbus_s	PVBus	Slave	-
reset_in	Signal	Slave	-
rvbar_cl0[1]	Value_64	Master	-
rvbar_cl1[2]	Value_64	Master	-
rvbar_cl2[4]	Value_64	Master	-

## Parameters for SI\_System\_Ctrl\_Regs

### **c10\_c0\_cfgrvbaraddr**

#### **Type**

int

#### **Default value**

0x120000000

#### **Description**

CL0\_C0\_CFGRVBARADDR.

### **c11\_c0\_cfgrvbaraddr**

#### **Type**

int

#### **Default value**

0x140000000

#### **Description**

CL1\_C0\_CFGRVBARADDR.

### **c11\_c1\_cfgrvbaraddr**

#### **Type**

int

#### **Default value**

0x140002000

#### **Description**

CL1\_C1\_CFGRVBARADDR.

### **c12\_c0\_cfgrvbaraddr**

#### **Type**

int

#### **Default value**

0x160000000

#### **Description**

CL2\_C0\_CFGRVBARADDR.

### **c12\_c1\_cfgrvbaraddr**

#### **Type**

int

#### **Default value**

0x160002000

**Description**

CL2\_C1\_CFGRVBARADDR.

**c12\_c2\_cfgrvbaraddr****Type**

int

**Default value**

0x160004000

**Description**

CL2\_C2\_CFGRVBARADDR.

**c12\_c3\_cfgrvbaraddr****Type**

int

**Default value**

0x160008000

**Description**

CL2\_C3\_CFGRVBARADDR.

**cpuhalt\_reset****Type**

int

**Default value**

0x0

**Description**

CPU HALT Reset value.

**diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

### 3.7.52 SMCF

System Monitoring Control Framework (SMCF). This model is written in C++.

#### Iris and MTI instances for SMCF

This model has the following Iris instances:

**Table 3-730: SMCF Iris instances**

InstanceName	ComponentName
SMCF	SMCF

#### Ports for SMCF

**Table 3-731: Ports**

Name	Protocol	Type	Description
mgi_clk_in	ClockSignal	Slave	Clock input
mgi_irq_out	Signal	Master	Interrupt signal output
mgi_pvbus_m	PVBus	Master	For DMA or memory mapped data write
mgi_reg_pvbus_s	PVBus	Slave	To access MGI register
mgi_reset_in	Signal	Slave	Reset signal input
mgi_tag_in[4]	Value	Slave	Tag value from external hardware
mgi_trigger_in	Signal	Slave	To trigger the start of a sample by external hardware
mgi_trigger_out	Signal	Master	Signal to external hardware to indicate that an event has occurred in an MGI
mli_hsp_enable_ack_in[32]	Signal	Slave	Acknowledgement signal for enable signal in hsp.
mli_hsp_enable_out[32]	Signal	Master	Signal to enable Ring Oscillators in hsp.
mli_powerdown_in[32]	Signal	Slave	It gives info about AP is power OFF or not
mli_temperature_in[32]	ValueState	Slave	It will be connected to temperature sensor to fetch the temperature value
smcf_mli_pvbus_m[32]	PVBus	Master	manager port to read/write the AMU or HSP register value

#### Parameters for SMCF

##### ALERT\_NUM\_CFG

###### Type

int

###### Default value

0x0

###### Description

Specifies the number of alerts present A value of 0 means that no alerts are present.

##### ALT\_ADDR\_CFG

###### Type

int



**Default value**

0x0

**Description**

Specifies where monitor data is read from, it is either: 0: The MGI\_DATA<n> registers. 1: The address specified in MGI\_RADDR0/1.

**ALT\_DELTA\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies the presence of alert rising and falling delta functions 0: The rising and falling delta functions are not present 1: The rising and falling delta functions are present.

**DATA\_PER\_MON\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies the number of data values(DATA\_PER\_MON\_CFG+1) generated from each monitor.

**DEF\_CFG\_IRQ\_MASK****Type**

int

**Default value**

0x1

**Description**

Specifies the default value of the configuration interrupt event mask. Sets the reset value of MGI\_IRQ\_MASK.CFG\_IRQ\_MASK.

**DEF\_CFG\_TRIG\_MASK****Type**

int

**Default value**

0x1

**Description**

Specifies the default value of the configuration trigger event mask. Sets the reset value of MGI\_TRG\_MASK.CFG\_TRIG\_MASK.

**DEF\_RADDR\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies the default alternate read address. This sets the default value for MGI\_RADDR0/MGI\_RADDR1.

**DEF\_WADDR\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies the default write address for the DMA interface. This sets the default value for MGI\_WADDR0/MGI\_WADDR1. Only required if DMA\_IF\_CFG = 1. This value must be 32-bit aligned.

**DMA\_IF\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies the presence of the DMA interface 0: DMA interface is not present 1: DMA interface is present.

**END\_COMPONENT****Type**

int

**Default value**

0x0

**Description**

Based on this parameter value sampling value will be fetched from the respective model 0: Temperature sensor 1: Monitor unit (AMU) 2: Fake sensor/monitor 3: HSP (Hot Spot Profiler) 4: DTSV2(Distributed Thermal Sensor V2) 5: DTS(Digital Distributed Temperature Sensor).

**FAKE\_SENSOR\_MAX\_LIMIT****Type**

int

**Default value**

0xffff

**Description**

Maximum value limit for the fake sensor/monitor.

**FAKE\_SENSOR\_MIN\_LIMIT****Type**

int

**Default value**

0x0

**Description**

Minimum value limit for the fake sensor/monitor.

**GRP\_ID\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies a unique identifier for an MGI.

**MGI\_AIDR\_RESET\_VALUE****Type**

int

**Default value**

0x10

**Description**

Reset value for MGI\_AIDR register.

**MGI\_IIDR\_RESET\_VALUE****Type**

int

**Default value**

0x8d00043b

**Description**

Reset value for MGI\_IIDR register.

**MLI\_QUANTITY****Type**

int

**Default value**

0x1

**Description**

Specifies the number of MLI.

**MODE\_LEN\_CFG****Type**

int

**Default value**

0x1f

**Description**

Specifies the bit width of each MGI\_MODE\_REQ/STAT. The number of bits is MODE\_LEN\_CFG+1.

**MODE\_REG\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies the number of MGI\_MODE\_REQ/STAT pairs. A value of 0 means that no MGI\_MODE\_REQ/STAT pairs are present.

**MONITOR\_SAMPLE\_TIME****Type**

int

**Default value**

0x13fff

**Description**

Time that monitor takes to perform the sampling (essentially the time from MGI sending the sample\_start command to the MGI receiving all the sample data).

**MON\_BASE\_ADDRESS****Type**

string

**Default value**

"0x00000000"

**Description**

base address used for sampling the sensors. It is used for configuration writes and data read bursts.

**MON\_DATA\_WIDTH\_CFG****Type**

int

**Default value**

0x1f

**Description**

Specifies the bit width(MON\_DATA\_WIDTH\_CFG+1) of each monitor data value.

**MON\_DISCON\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies if each monitor supports being disconnected. For example, bit 0 represents monitor 0, and bit 3 represents monitor 3. 0b0: Monitor does not support being disconnected. It is reset to connected. 0b1: Monitor supports being disconnected. It is reset to disconnected.

**MON\_NUM\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies the number of monitors(MON\_NUM\_CFG+1) in an MGI.

**NUM\_OF\_RSP\_CONNECTED****Type**

int

**Default value**

0x9

**Description**

Number of maximum RSPs can be connected to the DTSV2.

**PACKED\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies if monitor data is packed 0: Data is not packed 1: Data is packed.

**PER\_TIMER\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies the presence of the periodic timer 0: The periodic timer is not present 1: The periodic timer is present.

**SINGLE\_MON\_MODE\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies that each monitor is a single type and all monitors will have the same mode setting.

**SMP\_DLY\_LEN\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies the bit width of MGI\_SMP\_DLY. The number of bits in SMP\_DLY\_LEN\_CFG. A value of 0 means this register is not present and the feature is not supported.

**TAG\_IN\_CFG****Type**

int

**Default value**

0x1

**Description**

Specifies the presence of the tag input 0: The tag input is not present 1: The tag input is present.

**TAG\_LEN\_CFG****Type**

int

**Default value**

0x1f

**Description**

Specifies the bit width of the tag input The tag bit width is TAG\_LEN\_CFG+1.

**TRIG\_IN\_CFG****Type**

int

**Default value**

0x1

**Description**

Specifies the presence of the input trigger interface 0: The trigger in interface is not present 1: The trigger in interface is present.

**TRIG\_OUT\_CFG****Type**

int

**Default value**

0x1

**Description**

Specifies the presence of the output trigger interface 0: The trigger out interface is not present 1: The trigger out interface is present.

**USER\_DEF\_CMD\_CFG****Type**

int

**Default value**

0x0

**Description**

Specifies if an MGI supports User-Defined commands. 0: User-Defined commands are not supported. 1: User-Defined commands are supported.

**diagnostics****Type**

int

**Default value**

0x2

**Description**

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

### 3.7.53 SMMUv3TestEngine

Test Engine used for testing SMMUv3. This model is written in C++.

**Iris and MTI instances for SMMUv3TestEngine**

This model has the following Iris instances:

**Table 3-732: SMMUv3TestEngine Iris instances**

InstanceName	ComponentName
SMMUv3TestEngine	SMMUv3TestEngine
SMMUv3TestEngine.register_file[0]	PVBusSlave

This model has the following MTI trace components:

**Table 3-733: SMMUv3TestEngine MTI instances**

InstanceName	ComponentName
SMMUv3TestEngine	SMMUv3TestEngine
SMMUv3TestEngine.register_file[0]	PVBusSlave

**Ports for SMMUv3TestEngine****Table 3-734: Ports**

Name	Protocol	Type	Description
client_s	PCIDevice2ClientProtocol	Slave	-
clk_in	ClockSignal	Slave	-
identify	SMMUv3AEMIdentifyProtocol	Slave	-
pvbus_control_s	PVBus	Slave	-
pvbus_m[64]	PVBus	Master	-
reset_in	Signal	Slave	-



## Parameters for SMMUv3TestEngine

### **bandwidth\_per\_transaction\_in\_bytes\_per\_tick**

**Type**

int

**Default value**

0x64

**Description**

The bandwidth of the device for each in-flight transaction, in bytes/tick of clk\_in. This is only a rough guess. If you are uninterested in trying to run cores and the engine simultaneously then set this to a large number.

### **max\_number\_of\_inflight\_transactions**

**Type**

int

**Default value**

0xa

**Description**

The maximum number of in-flight transactions allowed.

### **output\_attribute\_transform**

**Type**

string

**Default value**

""

**Description**

How to pack the stream identification information into the transaction attributes. This is:- \* <empty> or "default" \* "pcie" the de-facto standard for the PCIe subsystem in FastModels The <empty> or "default" is equivalent to:- ExtendedID[63]=nSEC\_SID, ExtendedID[55:24]=StreamID, ExtendedID[20]=nSSV, ExtendedID[19:0]=SubstreamID The "pcie" option is equivalent to:- ExtendedID[63]=SEC\_SID, ExtendedID[62]=SSV, ExtendedID[51:32]=SubstreamID, ExtendedID[31:0]=StreamID .

### **seed**

**Type**

int

**Default value**

0x12345678

**Description**

The seed to use for initialising the random number generators.

### 3.7.54 STLBusGasket

STLBusGasket allows a debugger or emulated T32 code to force the results of system-register reads by writing an address to the ADDR register then 32-bit values to the VALUE register, which are placed in a fifo associated with that address. A PVBUS transaction into pvbus\_in goes unchanged to pvbus\_out, unless its address matches that associated with a non-empty fifo, in which case: writes are ignored, non-word reads abort, and word reads take values from that fifo. This model is written in LISA+.

#### Iris and MTI instances for STLBusGasket

This model has the following Iris instances:

Table 3-735: STLBusGasket Iris instances

InstanceName	ComponentName
STLBusGasket	STLBusGasket
STLBusGasket.busmapper	PVBusMapper

This model has the following MTI trace components:

Table 3-736: STLBusGasket MTI instances

InstanceName	ComponentName
STLBusGasket.busmapper	PVBusMapper

#### Ports for STLBusGasket

Table 3-737: Ports

Name	Protocol	Type	Description
pvbus_in	PVBus	Slave	-
pvbus_out	PVBus	Master	-

#### Parameters for STLBusGasket

**function**

Type

int

Default value

0x0

Description

Function: 0-none, 1-STL value-forcing.

**reg\_base**

Type

int

**Default value**

0xe001e820

**Description**

Base Address of STL control regs (ADDR,VAL at offsets 0,4).

**verbose**

**Type**

int

**Default value**

0x0

**Description**

Verbosity : 0-none, 1-some.

3.7.55 SerialCrossover

Implement an equivalent to a null-modem cable, swapping over serial transmit and receive signals. This model is written in C++.

**About SerialCrossover**

This component implements two SerialData slave ports and can connect two SerialData master ports, such as from PL011\_Uart components. Data received on one port is buffered in a FIFO until it is read from the other port. Signals received on one port are latched and available to be read by the other port.

**Ports for SerialCrossover**

Table 3-738: Ports

Name	Protocol	Type	Description
port_a	SerialData	Slave	Slave port for connecting to a SerialData master.
port_b	SerialData	Slave	Slave port for connecting to a SerialData master.

3.7.56 Signal\_Multiplexer

Signal Multiplexer. This model is written in C++.

**Iris and MTI instances for Signal\_Multiplexer**

This model has the following Iris instances:

Table 3-739: Signal\_Multiplexer Iris instances

InstanceName	ComponentName
Signal_Multiplexer	Signal_Multiplexer

## Ports for Signal\_Multiplexer

**Table 3-740: Ports**

Name	Protocol	Type	Description
input[16]	Signal	Slave	-
output	Signal	Master	-
selector	Value	Slave	-

## Parameters for Signal\_Multiplexer

### diagnostics

#### Type

int

#### Default value

0x2

#### Description

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

## 3.7.57 SoC\_SOR

System Override Registers unit. This model is written in C++.

### Iris and MTI instances for SoC\_SOR

This model has the following Iris instances:

**Table 3-741: SoC\_SOR Iris instances**

InstanceName	ComponentName
SoC_SOR	SoC_SOR
SoC_SOR.hdlcd0_override	PVBusMapper
SoC_SOR.hdlcd1_override	PVBusMapper
SoC_SOR.pvbusslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-742: SoC\_SOR MTI instances**

InstanceName	ComponentName
SoC_SOR.hdlcd0_override	PVBusMapper
SoC_SOR.hdlcd1_override	PVBusMapper
SoC_SOR.pvbusslave	PVBusSlave

Ports for SoC\_SOR

Table 3-743: Ports

Name	Protocol	Type	Description
hdlcd0_pvbus_s	PVBus	Slave	-
hdlcd1_pvbus_s	PVBus	Slave	-
hdlcd_pvbus_m	PVBus	Master	-
pvbus_s	PVBus	Slave	-

Parameters for SoC\_SOR

diagnostics

Type

int

Default value

0x0

Description

Diagnostics.

gpr0

Type

int

Default value

0x0

Description

General Purpose Register 0.

gpr1

Type

int

Default value

0x0

Description

General Purpose Register 1.

### 3.7.58 SoC\_Temperature

System On-Chip Temperature component which provides averaged out temperature of Application Processors. This model is written in C++.

#### Iris and MTI instances for SoC\_Temperature

This model has the following Iris instances:

**Table 3-744: SoC\_Temperature Iris instances**

InstanceName	ComponentName
SoC_Temperature	SoC_Temperature
SoC_Temperature.target	PVBusSlave

This model has the following MTI trace components:

**Table 3-745: SoC\_Temperature MTI instances**

InstanceName	ComponentName
SoC_Temperature.target	PVBusSlave

#### Ports for SoC\_Temperature

**Table 3-746: Ports**

Name	Protocol	Type	Description
pvbus_reg_s	PVBus	Slave	-
temperature_in[16]	ValueState	Slave	-

#### Parameters for SoC\_Temperature

##### diagnostics

###### Type

int

###### Default value

0x0

###### Description

Diagnostics.

##### num\_cores

###### Type

int

###### Default value

0x1

###### Description

Number of AP cores in the system.

### 3.7.59 SystemIdUnit

System ID Unit. This model is written in C++.

#### Iris and MTI instances for SystemIdUnit

This model has the following Iris instances:

Table 3-747: SystemIdUnit Iris instances

InstanceName	ComponentName
SystemIdUnit	SystemIdUnit
SystemIdUnit.pvbuslave	PVBusSlave

This model has the following MTI trace components:

Table 3-748: SystemIdUnit MTI instances

InstanceName	ComponentName
SystemIdUnit.pvbuslave	PVBusSlave

#### Ports for SystemIdUnit

Table 3-749: Ports

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	-

#### Parameters for SystemIdUnit

##### chip\_id

###### Type

int

###### Default value

0x0

###### Description

the ID for the node/chip when there are multiple SoCs.

##### chiplet\_type

###### Type

bool

###### Default value

0x0

###### Description

Specifies Compute Chiplet (CC) or Specialization Chiplet (SC). 0 - CC, 1 - SC.

**diagnostics****Type**

int

**Default value**

0x2

**Description**

Diagnostics.

**multi\_chip\_mode****Type**

bool

**Default value**

0x0

**Description**

Multi chip mode?.

**platform\_type****Type**

int

**Default value**

0x0

**Description**

the type of the subsystem: 0=mobile, 1=InfraSysDesign4.x, 2=InfraSysDesign5.x, 3=InfraSysDesign6.x, 4=SafetyIsland, 5=Client.

**soc\_id****Type**

int

**Default value**

0x0

**Description**

the ID for the SoC that integrates the subsystem.

**system\_cfg****Type**

int

**Default value**

0x0



**Description**

the ID for the subsystem configuration.

**system\_id****Type**

int

**Default value**

0x0

**Description**

the version ID for the subsystem.

### 3.7.60 TC25\_SecureAccessConfig

Secure Control Register Block for TC25. This model is written in C++.

#### Iris and MTI instances for TC25\_SecureAccessConfig

This model has the following Iris instances:

**Table 3-750: TC25\_SecureAccessConfig Iris instances**

InstanceName	ComponentName
TC25_SecureAccessConfig	TC25_SecureAccessConfig
TC25_SecureAccessConfig.bus_mapper	PVBusMapper
TC25_SecureAccessConfig.busslave_ns	PVBusSlave
TC25_SecureAccessConfig.busslave_s	PVBusSlave
TC25_SecureAccessConfig.idau_busmaster	PVBusMaster
TC25_SecureAccessConfig.p_ahb_bus_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-751: TC25\_SecureAccessConfig MTI instances**

InstanceName	ComponentName
TC25_SecureAccessConfig.bus_mapper	PVBusMapper
TC25_SecureAccessConfig.busslave_ns	PVBusSlave
TC25_SecureAccessConfig.busslave_s	PVBusSlave
TC25_SecureAccessConfig.idau_busmaster	PVBusMaster
TC25_SecureAccessConfig.p_ahb_bus_mapper	PVBusMapper

#### Ports for TC25\_SecureAccessConfig

**Table 3-752: Ports**

Name	Protocol	Type	Description
acc_waitn	ValueState	Master	-

Name	Protocol	Type	Description
brg_in[32]	StateSignal	Slave	-
brg_out	Signal	Master	-
idau	PVBus	Master	-
mainnsppcexp[4]	ValueState	Master	-
mainpppcexp[4]	ValueState	Master	-
mem_gating_filter_in	PVBus	Slave	-
mem_gating_filter_out	PVBus	Master	-
mpc_in[32]	StateSignal	Slave	-
mpc_out	Signal	Master	-
msc_in[32]	StateSignal	Slave	-
msc_out	Signal	Master	-
npuspporpl[4]	Signal	Master	-
npuspporsl[4]	Signal	Master	-
p_ahb_gating_filter_in	PVBus	Slave	-
p_ahb_gating_filter_out	PVBus	Master	-
periphnsppc0	ValueState	Master	-
periphnsppc1	ValueState	Master	-
periphnsppcexp[4]	ValueState	Master	-
periphpppc0	ValueState	Master	-
periphpppc1	ValueState	Master	-
periphpppcexp[4]	ValueState	Master	-
ppc_in[32]	StateSignal	Slave	-
ppc_out	Signal	Master	-
pvbus_nonsecure	PVBus	Slave	-
pvbus_secure	PVBus	Slave	-
reset_in	Signal	Slave	-
security_resp	ValueState	Master	-

## Parameters for TC25\_SecureAccessConfig

### CODENSC

#### Type

bool

#### Default value

0x0

#### Description

Whether 0x10000000..0x1FFFFFFF is non-secure-callable.

**DISABLE\_GATING****Type**

bool

**Default value**

0x0

**Description**

Disable Memory gating logic.

**IGNORE\_MEM\_MAP****Type**

bool

**Default value**

0x0

**Description**

Ignore Memory mapping logic.

**MAINPPCEXP\_DIS0****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the MAINNSPPCEXP0 and MAINPPPCEXP0 buses.

**MAINPPCEXP\_DIS1****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the MAINNSPPCEXP1 and MAINPPPCEXP1 buses.

**MAINPPCEXP\_DIS2****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the MAINNSPPCEXP2 and MAINPPPCEXP2 buses.

**MAINPPCEXP\_DIS3****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the MAINNSPPCEXP3 and MAINPPPCEXP3 buses.

**PERIPHPPCEXP\_DIS0****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the PERIPHNSPPCEXP0 and PERIPHPPPCEXP0 buses.

**PERIPHPPCEXP\_DIS1****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the PERIPHNSPPCEXP1 and PERIPHPPPCEXP1 buses.

**PERIPHPPCEXP\_DIS2****Type**

int

**Default value**

0x0

**Description**

Disables support for individual bits on the PERIPHNSPPCEXP2 and PERIPHPPPCEXP2 buses.

**PERIPHPPCEXP\_DIS3****Type**

int

**Default value**

0x0

**Description**  
Disables support for individual bits on the PERIPHNSPPCEXP3 and PERIPHPPPCEXP3 buses.

**RAMNSC**

**Type**  
bool

**Default value**  
0x0

**Description**  
Whether 0x30000000..0x3FFFFFFF is non-secure-callable.

**diagnostics**

**Type**  
int

**Default value**  
0x0

**Description**  
Diagnostics.

3.7.61 TRNG

True Random Number Generator. This model is written in C++.

Iris and MTI instances for TRNG

This model has the following Iris instances:

Table 3-753: TRNG Iris instances

InstanceName	ComponentName
TRNG	TRNG
TRNG.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Table 3-754: TRNG MTI instances

InstanceName	ComponentName
TRNG.pvbusslave	PVBusSlave

## Ports for TRNG

**Table 3-755: Ports**

Name	Protocol	Type	Description
cc_host_int_req	Signal	Master	-
pvbus_s	PVBus	Slave	-
rng_clk	ClockSignal	Slave	-
rst_n	Signal	Slave	-
scanmode	Signal	Slave	-

## Parameters for TRNG

### diagnostics

#### Type

int

#### Default value

0x0

#### Description

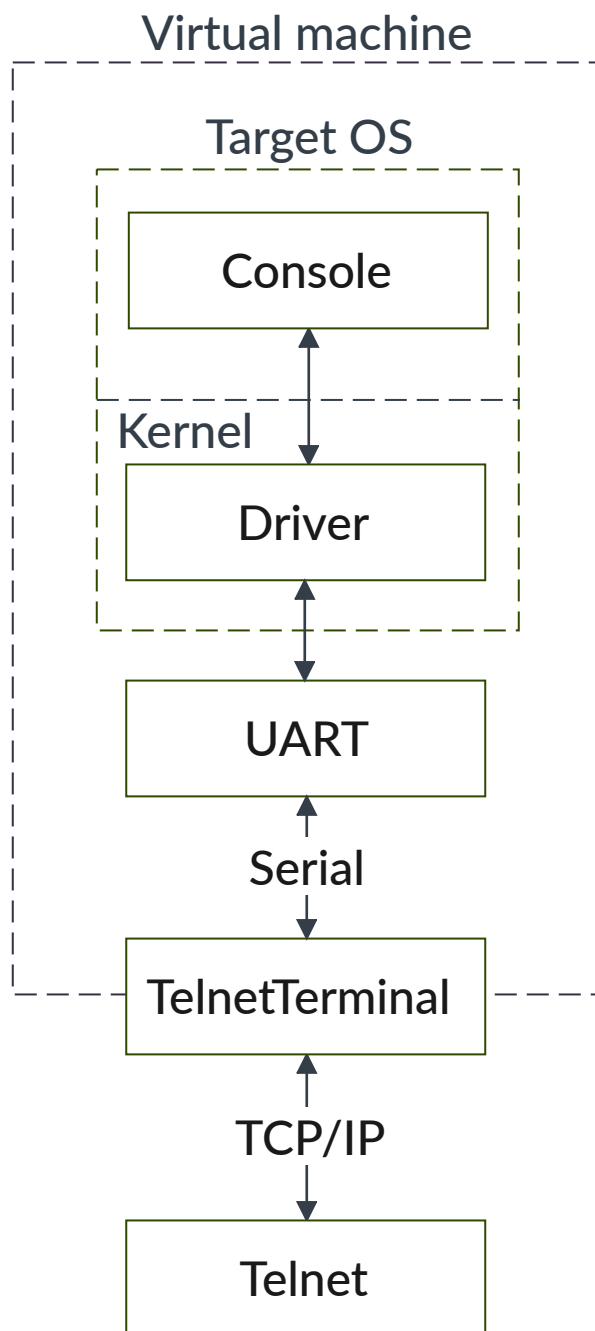
Diagnostics.

## 3.7.62 TelnetTerminal

A host interface onto a serial port: exposes the two way serial data channel over a TCP/IP interface, and automatically opens a telnet application connected to the network socket, unless a user application connects first. This model is written in C++.

### Using TelnetTerminal

The following figure shows a block diagram of one possible relationship between the target and host through the TelnetTerminal component. The TelnetTerminal block is what you configure when you define Terminal component parameters. The Virtual Machine is an FVP.

**Figure 3-2: Terminal block diagram**

On the target side, the console process that is invoked by your target OS relies on a suitable driver being present. Such drivers are normally part of the OS kernel. The driver passes serial data

through a UART, for example [3.10.74 PL011\\_Uart](#) on page 4909. The data is forwarded to the TelnetTerminal component. When the simulation is started and the TelnetTerminal component is enabled, the component opens a server (listening) socket on a TCP/IP port. This is port 5000 by default. This port can be connected to by, for example, a Telnet process on the host.

When data becomes available on the network socket, the TelnetTerminal component buffers the data, which can then be read from SerialData.

If there is no connection to the network socket when the first data access is made, and the `start_telnet` parameter is true, a host Telnet session is started automatically. Prior to this first access, you can connect a client of your choice to the network socket.

If the connection between the TelnetTerminal component and the client is broken at any time, for example by closing a client Telnet session, the port is re-opened on the host, permitting you to make another client connection. This could have a different port number if the original one is no longer available.

The port number of a particular TelnetTerminal instance can be defined when your model system starts. The actual value of the port used by each TelnetTerminal is declared when it starts or restarts, and might not be the value that you specified if the port is already in use. If you are using Model Shell, the port numbers are displayed in the host window in which you started the model.



Note

Microsoft Windows 10 disables the Telnet client by default. Follow these steps to enable it:

1. Select **Start > Settings**.
2. In the search box, type **Turn Windows features on or off**. The **Windows Features** dialog opens.
3. Select the **Telnet Client** check box and click **OK**. The installation might take several minutes to complete.

## TelnetTerminal parameters

To set the parameters, the syntax to use in a configuration file or on the command line is:

```
motherboard.terminal_x.parameter=value
```

where *x* is the terminal identifier and can be 0, 1, 2, or 3.

You can start the TelnetTerminal component in either of the following modes, depending on the `mode` parameter:

### telnet

In Telnet mode, the terminal component supports a subset of the RFC 854 protocol. This means that the terminal participates in negotiations between the host and client concerning what is and is not supported, but there is no flow control.



**raw**

In raw mode the byte stream passes unmodified between the host and the target. The terminal does not participate in initial capability negotiations between the host and client. Instead it acts as a TCP/IP port. You can use this feature to directly connect to your target through the TelnetTerminal component. This permits a debugger connection, for example, to connect a gdb client to a gdbserver running on the target operating system.

The `terminal_command` parameter specifies the command line used to launch a terminal application and connect to the opened TCP port. The TelnetTerminal component replaces the keywords `%port` and `%title`, if specified, with the opened port number and component name, respectively. After replacing `%port` and `%title`, the command line is executed verbatim.

An empty string, which is the default, launches `xterm` on Linux or `telnet.exe` on Windows.



**Note**

If you specify a non-empty string, it must include `%port`, but `%title` is optional.

For example:

```
fvp_mps2.telnetterminal0.terminal_command="putty.exe -telnet localhost %port"
```

**Iris and MTI instances for TelnetTerminal**

This model has the following Iris instances:

**Table 3-756: TelnetTerminal Iris instances**

InstanceName	ComponentName
TelnetTerminal	TelnetTerminal

**Ports for TelnetTerminal**

**Table 3-757: Ports**

Name	Protocol	Type	Description
serial	SerialData	Slave	Slave port for connecting to a SerialData master.

**Parameters for TelnetTerminal**

**mode**

**Type**

string

**Default value**

"telnet"

**Description**

Terminal initialisation mode.

**quiet****Type**

bool

**Default value**

0x0

**Description**

Avoid output on stdout/stderr.

**start\_port****Type**

int

**Default value**

0x1388

**Description**

Telnet TCP Port Number.

**start\_telnet****Type**

bool

**Default value**

0x1

**Description**

Start telnet if nothing connected.

**terminal\_command****Type**

string

**Default value**

""

**Description**

Commandline to launch a terminal application and connect to the opened TCP port. Keywords %port and %title will be replaced with the opened port number and component name respectively. An empty string (default behaviour) will launch xterm (Linux) or telnet.exe (Windows).

### 3.7.63 Temperature

Component to synthesis the temperature value of the connected core. This model is written in C++.

#### Iris and MTI instances for Temperature

This model has the following Iris instances:

**Table 3-758: Temperature Iris instances**

InstanceName	ComponentName
Temperature	Temperature

#### Ports for Temperature

**Table 3-759: Ports**

Name	Protocol	Type	Description
cluster_powerdown	Signal	Slave	-
core_powerdown[8]	Signal	Slave	-
freq_changed	ValueState	Slave	-
temperature	ValueState	Master	-

#### Parameters for Temperature

##### **core\_coefficient**

###### Type

int

###### Default value

0x0

###### Description

Temperature Coefficient.

##### **diagnostics**

###### Type

int

###### Default value

0x0

###### Description

Diagnostics.

##### **num\_cores**

###### Type

int

**Default value**

0x4

**Description**

Number of cores.

3.7.64 TestbedGPIOConnector

Tool for receiving GPIO signals and reporting test success/failure. This model is written in C++.

**Iris and MTI instances for TestbedGPIOConnector**

This model has the following Iris instances:

Table 3-760: TestbedGPIOConnector Iris instances

InstanceName	ComponentName
TestbedGPIOConnector	TestbedGPIOConnector

Ports for TestbedGPIOConnector

Table 3-761: Ports

Name	Protocol	Type	Description
from_gpio0	Value	Slave	-
from_gpio1	Value	Slave	-

Parameters for TestbedGPIOConnector

**active**

**Type**

bool

**Default value**

0x1

**Description**

Actually stop the simulator when testbench running in it reports results over GPIO.

**diagnostics**

**Type**

int

**Default value**

0x0

**Description**

Diagnostic level.

### 3.7.65 UnusedPrimeCell

A dummy component. It can be used to represent any unimplemented PrimeCell components. This model is written in LISA+.

#### Iris and MTI instances for UnusedPrimeCell

This model has the following Iris instances:

**Table 3-762: UnusedPrimeCell Iris instances**

InstanceName	ComponentName
UnusedPrimeCell	UnusedPrimeCell
UnusedPrimeCell.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-763: UnusedPrimeCell MTI instances**

InstanceName	ComponentName
UnusedPrimeCell.busslave	PVBusSlave

#### Ports for UnusedPrimeCell

**Table 3-764: Ports**

Name	Protocol	Type	Description
pvbus	PVBus	Slave	Bus slave interface.

### 3.7.66 Value\_Multiplexer

Signal Multiplexer. This model is written in C++.

#### Iris and MTI instances for Value\_Multiplexer

This model has the following Iris instances:

**Table 3-765: Value\_Multiplexer Iris instances**

InstanceName	ComponentName
Value_Multiplexer	Value_Multiplexer

#### Ports for Value\_Multiplexer

**Table 3-766: Ports**

Name	Protocol	Type	Description
input[16]	Value	Slave	-
output	Value	Master	-
selector	Value	Slave	-

## Parameters for Value\_Multiplexer

### diagnostics

#### Type

int

#### Default value

0x2

#### Description

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

## 3.7.67 VirtioBlockDevice

virtio block device. This model is written in C++.

### About VirtioBlockDevice

VirtioBlockDevice implements a block device that can be accessed from the simulated OS if it has an appropriate driver. Similarly to the VirtioP9Device, this component is targeted primarily at Linux, which has a built-in virtio block driver. VirtioBlockDevice allows you to use a file on the host that you specify using the `image_path` parameter, as a hard drive in the simulated OS.

VirtioBlockDevice supports the legacy OASIS virtio specification.

Unlike the VirtioP9Device, you should not need to carry out any special setup to use VirtioBlockDevice on VE or Base platforms, because it is usually already included in the device trees. Set the `image_path` parameter to point to your image, and then on Linux it is available as a block device, usually `/dev/vda`, which you then use like any other hard drive.

### Iris and MTI instances for VirtioBlockDevice

This model has the following Iris instances:

**Table 3-767: VirtioBlockDevice Iris instances**

InstanceName	ComponentName
VirtioBlockDevice	VirtioBlockDevice
VirtioBlockDevice.register_slave	PVBusSlave
VirtioBlockDevice.virtio_master	PVBusMaster

This model has the following MTI trace components:

**Table 3-768: VirtioBlockDevice MTI instances**

InstanceName	ComponentName
VirtioBlockDevice	VirtioBlockDevice
VirtioBlockDevice.register_slave	PVBusSlave
VirtioBlockDevice.virtio_master	PVBusMaster

Ports for VirtioBlockDevice

Table 3-769: Ports

Name	Protocol	Type	Description
intr	Signal	Master	Virtio device sets interrupt to signal completion.
pvbus	PVBus	Slave	Virtio MMIO control/config/status registers.
virtio_m	PVBus	Master	Virtio device performs DMA accesses via master.

Parameters for VirtioBlockDevice

**image\_path**

Type

string

Default value

""

Description

image file path.

**quiet**

Type

bool

Default value

0x0

Description

Don't print warnings on malformed commands/descriptors.

**read\_only**

Type

bool

Default value

0x0

Description

Only allow device to be read.

**secure\_accesses**

Type

bool

Default value

0x0

Description

Make device generate transactions with NS=0.

**transaction\_attributes**

**Type**

int

**Default value**

0x0

**Description**

Transaction attributes used by device. 0x0 - inner-shared real access. 0x1 - outer-shared real access. 0x2 - outer-shared debug access.

3.7.68 VirtioBlockDeviceMMIO

Virtio v1.0 Block device over MMIO transport. This model is written in C++.

About VirtioBlockDeviceMMIO

VirtioBlockDeviceMMIO supports both the legacy and v1.0 OASIS virtio specifications.



VirtioBlockDeviceMMIO is an evolution of [VirtioBlockDevice](#), which is also MMIO-based and has the same ports, but only supports the legacy OASIS virtio specification.

Iris and MTI instances for VirtioBlockDeviceMMIO

This model has the following Iris instances:

Table 3-770: VirtioBlockDeviceMMIO Iris instances

InstanceName	ComponentName
VirtioBlockDeviceMMIO	VirtioBlockMMIO
VirtioBlockDeviceMMIO.dma_master	PVBusMaster

This model has the following MTI trace components:

Table 3-771: VirtioBlockDeviceMMIO MTI instances

InstanceName	ComponentName
VirtioBlockDeviceMMIO.dma_master	PVBusMaster

Ports for VirtioBlockDeviceMMIO

Table 3-772: Ports

Name	Protocol	Type	Description
intr	Signal	Master	Virtio device sets interrupt to signal completion.
pvbus	PVBus	Slave	Virtio MMIO control/config/status registers.
virtio_m	PVBus	Master	Virtio device performs DMA accesses via master.



## Parameters for VirtioBlockDeviceMMIO

### **enabled**

#### Type

bool

#### Default value

0x0

#### Description

Enable or disable device. If disabled, device can be accessed, but will not be activated.

### **image\_path**

#### Type

string

#### Default value

""

#### Description

Image file path.

### **quiet**

#### Type

bool

#### Default value

0x0

#### Description

Don't print info or warnings (e.g. on malformed commands/descriptors).

### **read\_only**

#### Type

bool

#### Default value

0x0

#### Description

Only allow device to be read. If that parameter is set to false and the image file cannot be opened in RW mode, the model will try to work around it by opening the file in RO mode.

### **secure\_accesses**

#### Type

bool

#### Default value

0x0

**Description**

Make device generate transactions with NS=0.

**transaction\_attributes****Type**

int

**Default value**

0x0

**Description**

Transaction attributes used by device. 0x0 - inner-shared real access. 0x1 - outer-shared real access. 0x2 - outer-shared debug access.

**transport****Type**

string

**Default value**

"modern"

**Description**

Choose legacy or modern virtio transport, if not specified, modern transport is used.

### 3.7.69 VirtioNetMMIO

Virtio net device over MMIO transport. This model is written in C++.

**About VirtioNetMMIO**

This is a model of a virtual Ethernet virtio device over MMIO transport, supporting both the legacy and v1.0 OASIS virtio specifications. It provides much better network performance than the SMSC\_91C111 component, because it features host-assisted network acceleration. This means that it can offload packet processing operations from the simulated OS on the target, to the host side. These operations include:

- Checksum computation
- TX packet segmentation
- RX packet combination

If the target simulated Linux or Linux-derived OS has an appropriate virtio net driver, Arm recommends you use VirtioNetMMIO instead of SMSC\_91C111.

Unlike SMSC\_91C111, which must work with an external HostBridge component, VirtioNetMMIO has a built-in HostBridge sub-component. The parameters to control the HostBridge are described in the VirtioNetMMIO parameters table, with the `hostbridge` parameter sub-namespace.

To enable tracing of user-mode networking, which can help to debug networking issues, set the `FASTSIM_USERNET_DUMP` environment variable to any or all of the following values:

```
arpin,arpout,udpin,udpout,etherin,etherout,ipv4in,ipv4out,
ipv4fragin,ipv4fragout,tcpin,tcpout,dhcpv4in,dhcpv4out
```

Take the following steps to set up this component in a virtual platform:

- Use a version of Linux that contains a virtio network driver.
- Add the following option to the Linux kernel configuration:

```
CONFIG_VIRTIO_NET=y
```

- Update the device tree to include the VirtioNetMMIO component, or specify it on the kernel command line, for example

```
virtio_mmio.device=0x10000@0x1c150000:76
```

The address range for both VE and Base platforms is 0x1C150000-0x1C15FFFF. The interrupt number is 44, or IRQ 76, for both VE and Base platforms.

- Select the hostbridge that you want to use to communicate with the host in the model:

```
virtio_net.hostbridge.userNetworking=true/false (User mode or TAP/TUN networking)
```

- Configure the networking environment, as described in [Configuring the networking environment for Linux](#).

## Example entries for DTS files

- Add this entry next to the corresponding `virtio_block` or `virtio_p9` entry:

```
virtio_net@0150000 {
    compatible = "virtio,mmio";
    reg = <0x150000 0x1000>;
    interrupts = <0x2c>;
};
```

- Add this entry to the interrupt map:

```
<0 0 44 &gic 0 44 4>;
```

## See also

- [Configuring the networking environment for Linux](#)

## Iris and MTI instances for VirtioNetMMIO

This model has the following Iris instances:

**Table 3-773: VirtioNetMMIO Iris instances**

InstanceName	ComponentName
VirtioNetMMIO	VirtioNetMMIO
VirtioNetMMIO.dma_master	PVBusMaster
VirtioNetMMIO.hostbridge	HostBridge

This model has the following MTI trace components:

**Table 3-774: VirtioNetMMIO MTI instances**

InstanceName	ComponentName
VirtioNetMMIO.dma_master	PVBusMaster

## Ports for VirtioNetMMIO

**Table 3-775: Ports**

Name	Protocol	Type	Description
intr	Signal	Master	Virtio device sets interrupt to signal completion.
pvbus	PVBus	Slave	Virtio MMIO control/config/status registers.
virtio_m	PVBus	Master	Virtio device performs DMA accesses via master.

## Parameters for VirtioNetMMIO

### checksum

#### Type

string

#### Default value

""

#### Description

For checksum-offloaded packets, if 'tx' is specified, outgoings will be checksummed by VirtioNet device; 'rx' is specified for incomings; 'all' for both.

### enabled

#### Type

bool

#### Default value

0x1

#### Description

Enable or disable device. If disabled, device can be accessed, but will not be activated.

### hostbridge.interfaceName

#### Type

string

**Default value**

""

**Description**

Host Interface.

**hostbridge.userNetOptions****Type**

string

**Default value**

""

**Description**

Control options for UserNet TCP/IP (for internal use only, please do not use).

**hostbridge.userNetPorts****Type**

string

**Default value**

""

**Description**

Listening ports to expose in user-mode networking.

**hostbridge.userNetSubnet****Type**

string

**Default value**

"172.20.51.0/24"

**Description**

Virtual subnet for user-mode networking.

**hostbridge.userNetworking****Type**

bool

**Default value**

0x1

**Description**

Enable user-mode networking.

**mac\_address****Type**

string

**Default value**

""

**Description**

Device MAC address, if not specified, a random MAC address is generated.

**offload****Type**

string

**Default value**

"gso, gro"

**Description**

Offload TCP/UDP segmentation/receiving operations to host.

**secure\_accesses****Type**

bool

**Default value**

0x0

**Description**

Make device generate transactions with NS=0.

**transport****Type**

string

**Default value**

"modern"

**Description**

Choose legacy or modern virtio transport, if not specified, modern transport is used.

### 3.7.70 VirtioP9Device

virtio P9 server. This model is written in C++.

**About VirtioP9Device**

This component implements a subset of the Plan 9 file protocol over a virtio transport. It enables you to access a directory on the host's filesystem within Linux, or another operating system that implements the protocol, running on a platform model.

It supports the legacy OASIS virtio specification.

## Limitations

VirtioP9Device implements a subset of the Linux 9P2000.L protocol. It has the following limitations:

- You can mount only one host directory per instance of the component.
- It supports a subset of 9P2000.L message types:
  - Tversion
  - Tlopen
  - Tlcreate
  - Tgetattr
  - Tsetattr
  - Treaddir
  - Tmkdir
  - Tattach
  - Twalk
  - Tread
  - Twrite
  - Tclunk
  - Tremove
  - Trename
  - On Linux hosts, it also supports Treadlink and Tsymlink.
- On Windows hosts:
  - It ignores Unix permissions when writing files.
  - It performs a simple mapping from Windows to Unix permissions when reading.
  - Symbolic links appear as regular files.
  - It does not perform writing, deleting, or renaming operations on a file that another process has open.

## Setting up the VirtioP9Device component

Take the following steps to set up this component:

- Use a version of Linux that supports v9fs over virtio and virtio-mmio devices.
- Update the device tree to include the VirtioP9Device component, or specify it on the kernel command-line, as shown below. The address range for both VE and Base platforms is 0x1C140000-0x1C14FFFF. The interrupt number is 43, or IRQ 75, for both VE and Base platforms.
- Set the following parameter to the directory on the host that you want to mount in the model:

### VE

```
motherboard.virtiop9device.root_path
```

**Base**

```
bp.virtio_p9device.root_path
```

- On Linux, mount the host directory by using the following command in the model:

```
$ mount -t 9p -o trans=virtio,version=9p2000.L FM <mount point>
```

Example kernel command-line argument:

```
virtio_mmio.device=0x10000@0x1c140000:75
```

Example entry for DTS files, to add next to the corresponding virtio\_block entry:

```
virtio_p9@0140000 {
    compatible = "virtio,mmio";
    reg = <0x0 0x1c140000 0x0 0x1000>;
    interrupts = <0x0 0x2b 0x4>;
};
```

**Iris and MTI instances for VirtioP9Device**

This model has the following Iris instances:

**Table 3-776: VirtioP9Device Iris instances**

InstanceName	ComponentName
VirtioP9Device	VirtioP9Device
VirtioP9Device.mmio_slave	PVBusSlave
VirtioP9Device.virtio_master	PVBusMaster

This model has the following MTI trace components:

**Table 3-777: VirtioP9Device MTI instances**

InstanceName	ComponentName
VirtioP9Device.mmio_slave	PVBusSlave
VirtioP9Device.virtio_master	PVBusMaster

**Ports for VirtioP9Device****Table 3-778: Ports**

Name	Protocol	Type	Description
intr	Signal	Master	Virtio device sets interrupt to signal completion.
pvbus	PVBus	Slave	Virtio MMIO control/config/status registers.
virtio_m	PVBus	Master	Virtio device performs DMA accesses via master.



## Parameters for VirtioP9Device

### **mount\_tag**

#### **Type**

string

#### **Default value**

"FM"

#### **Description**

mount tag.

### **quiet**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Don't print warnings on malformed commands/descriptors.

### **root\_path**

#### **Type**

string

#### **Default value**

""

#### **Description**

root directory path.

### **secure\_accesses**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Make device generate transactions with NS=0.

### 3.7.71 VirtioPCIBlockDevice

virtio block device with PCI transport. This model is written in C++.

#### About VirtioPCIBlockDevice

VirtioPCIBlockDevice is similar to VirtioBlockDevice, except it is PCI-based instead of MMIO-based. It supports the legacy OASIS virtio specification.

This device requires:

- The following BARs in rising order:
  - A Bar of 4K for Config accesses
  - A Bar of 4K for the MSI-X table
  - A Bar of 4K for the MSI-X PBA
- Conventional interrupts to be supported
- An `msix_table_size` of 2

#### Iris and MTI instances for VirtioPCIBlockDevice

This model has the following Iris instances:

**Table 3-779: VirtioPCIBlockDevice Iris instances**

InstanceName	ComponentName
VirtioPCIBlockDevice	VirtioPCIBlockDevice
VirtioPCIBlockDevice.register_slave	PVBusSlave
VirtioPCIBlockDevice.virtio_master	PVBusMaster

This model has the following MTI trace components:

**Table 3-780: VirtioPCIBlockDevice MTI instances**

InstanceName	ComponentName
VirtioPCIBlockDevice	VirtioPCIBlockDevice
VirtioPCIBlockDevice.register_slave	PVBusSlave
VirtioPCIBlockDevice.virtio_master	PVBusMaster

#### Ports for VirtioPCIBlockDevice

**Table 3-781: Ports**

Name	Protocol	Type	Description
client_s	PCIDevice2ClientProtocol	Slave	Interrupts for MSI-X table entries.
intr	Signal	Master	Virtio device sets interrupt to signal completion.
pvbus	PVBus	Slave	Virtio pci/control/config/status registers.
virtio_m	PVBus	Master	Virtio device performs DMA accesses via master.

## Parameters for VirtioPCIBlockDevice

### **image\_path**

#### Type

string

#### Default value

""

#### Description

image file path.

### **quiet**

#### Type

bool

#### Default value

0x0

#### Description

Don't print warnings on malformed commands/descriptors.

### **read\_only**

#### Type

bool

#### Default value

0x0

#### Description

Only allow device to be read.

### **secure\_accesses**

#### Type

bool

#### Default value

0x0

#### Description

Make device generate transactions with NS=0.

### **transaction\_attributes**

#### Type

int

#### Default value

0x0

## Description

Transaction attributes used by device. 0x0 - inner-shared real access. 0x1 - outer-shared real access. 0x2 - outer-shared debug access.

## 3.7.72 VirtioRNG

virtio rng - Entropy device. This model is written in C++.

### About VirtioRNG

VirtioRNG models a virtio entropy device as defined in the [Virtio 1.0 Specification](#). A virtual platform might need to integrate a VirtioRNG component to generate random numbers when:

- Linux or Android needs to generate kernel entropy. Hardware might do this using a timer, but this is not possible in the model because timers are not updated quickly enough.
- Security features are required, such as ssh.

### Integrate VirtioRNG into a platform

Integrate the VirtioRNG component by instantiating it in your board's LISA file and connecting it to the SoC virtio master bus and interrupt signal as follows:

```
// Instantiate components
composition {
...
    virtio_rng : VirtioRNG();
...
}

connection {
...
    // Find a suitable address space and connect it to the SoC's virtio_m bus
    busdecoder.pvbus_m_range[0x001C190000..0x001C19ffff] => virtio_rng.pvbus;
    virtio_rng.virtio_m => self.virtio_m;

    // Connect the IRQ to the GIC IRQ
    virtio_rng.intr => gic400.irqs[101];
...
}
```

To configure Linux or Android for VirtioRNG, use the following build parameters:

#### Linux:

- CONFIG\_VIRTIO\_MMIO=y
- CONFIG\_HW\_RANDOM=y
- CONFIG\_HW\_RANDOM\_VIRTIO=y

#### Android:

- --enable CONFIG\_VIRTIO\_MMIO
- --enable CONFIG\_HW\_RANDOM
- --enable CONFIG\_HW\_RANDOM\_VIRTIO

Use the following device tree parameters:

```
virtio_rng@1c190000 {
    compatible = "virtio,mmio";
    reg = <0x0 0x1c190000 0x0 0x200>;
    interrupts = <GIC_SPI 101 IRQ_TYPE_LEVEL_HIGH>;
};
```

Configure VirtioRNG using model parameters, for example:

```
-C "board.virtio_rng.enabled=1" \
-C "board.virtio_rng.seed=0" \
-C "board.virtio_rng.generator=2" \
-C "board.virtio_rng.diagnostics=4" \ # Optional
```

Use the following guest command line to test the integration:

```
// Generate random numbers
console/> cat /dev/hwrng
```

Iris and MTI instances for VirtioRNG

This model has the following Iris instances:

Table 3-782: VirtioRNG Iris instances

InstanceName	ComponentName
VirtioRNG	VirtioEntropyMMIO
VirtioRNG.dma_master	PVBusMaster

This model has the following MTI trace components:

Table 3-783: VirtioRNG MTI instances

InstanceName	ComponentName
VirtioRNG.dma_master	PVBusMaster

Ports for VirtioRNG

Table 3-784: Ports

Name	Protocol	Type	Description
intr	Signal	Master	Virtio device sets interrupt to signal completion.
pvbus	PVBus	Slave	Virtio control/config/status registers.
virtio_m	PVBus	Master	Virtio device performs DMA accesses via master.

Parameters for VirtioRNG

diagnostics

Type

int

**Default value**

0x0

**Description**

Prints debug information: 0 = disabled; 1 = generated seed and device; 4 = generated seed, device and generated numbers.

**enabled****Type**

bool

**Default value**

0x0

**Description**

Enable or disable device. If disabled, device can be accessed, but will not be activated.

**generator****Type**

int

**Default value**

0x0

**Description**

User-defined generator: 0 = xorshiftstar; 1 = rand48; 2 = mersenne.

**secure\_accesses****Type**

bool

**Default value**

0x0

**Description**

Make device generate transactions with NS=0.

**seed****Type**

int

**Default value**

0x0

**Description**

User-defined seed: 0 = uses a random seed; > 0 = user-defined fixed seed value.

**transport****Type**

string

**Default value**

"modern"

**Description**

Choose legacy or modern virtio transport, if not specified, modern transport is used.

### 3.7.73 VirtualEthernetCrossover

Ethernet Crossover Cable. This model is written in LISA+.

**About VirtualEthernetCrossover**

This component implements two VirtualEthernet slave ports and enables you to connect two VirtualEthernet master ports. It forwards data received on one port to the other port without delay.

**Iris and MTI instances for VirtualEthernetCrossover**

This model has the following Iris instances:

**Table 3-785: VirtualEthernetCrossover Iris instances**

InstanceName	ComponentName
VirtualEthernetCrossover	VirtualEthernetCrossover

This model has the following MTI trace components:

**Table 3-786: VirtualEthernetCrossover MTI instances**

InstanceName	ComponentName
VirtualEthernetCrossover	VirtualEthernetCrossover

**Ports for VirtualEthernetCrossover****Table 3-787: Ports**

Name	Protocol	Type	Description
deva	VirtualEthernet	Slave	Slave port for connecting to a VirtualEthernet master.
devb	VirtualEthernet	Slave	Slave port for connecting to a VirtualEthernet master.

### 3.7.74 VirtualEthernetHub3

3 Port Ethernet Hub. This model is written in LISA+.

**Iris and MTI instances for VirtualEthernetHub3**

This model has the following Iris instances:

**Table 3-788: VirtualEthernetHub3 Iris instances**

InstanceName	ComponentName
VirtualEthernetHub3	VirtualEthernetHub3

### Ports for VirtualEthernetHub3

**Table 3-789: Ports**

Name	Protocol	Type	Description
deva	VirtualEthernet	Slave	Slave port for connecting to a VirtualEthernet master.
devb	VirtualEthernet	Slave	Slave port for connecting to a VirtualEthernet master.
devc	VirtualEthernet	Slave	Slave port for connecting to a VirtualEthernet master.

## 3.7.75 VisEventRecorder

Event recorder component for visualisation component. It allows you to play back and record interactive GUI sessions. This model is written in LISA+.

### Recording

The following command creates an ASCII file `rec.txt` and enables recording. This file can directly be used for playback.

```
./isim_system -a image.axf -C visualisation.recorder.recordingFileName=rec.txt
```

You can select the time base for the time stamps of the recorded events. The default is a 100MHz counter (10ns simulated time resolution) which usually works for all systems. To be able to correlate timestamps to the instruction counter, set the time base to the clock frequency of the CPU, but this is not necessary for an exact recording or playback. The time base should be higher than CPU frequency / 100 (typical quantum size). To set the recording time base set the `recordingTimeBase` parameter.

### Playback

The following command enables the playback of all GUI input events previously recorded in file `rec.txt`. The time base of the timestamps is always taken from the file (see T record). Interactive user input is still possible and interactive events and recorded events are mixed.

```
./isim_system -a image.axf -C visualisation.recorder.playbackFileName=rec.txt
```



**Note**

It is possible to enable recording and playback at the same time. This makes it possible to check whether a playback session is reproducible or to extend a previously recorded session by appending events. To do this, remove the QUIT



event at the end. This is also useful to check the timing accuracy of the playback/recording timestamps.

```
./isim system -a image.axf -C
visualisation.recorder.playbackFileName=rec.txt -C
visualisation.recorder.playbackFileName=rec.txt
```

To enable verbose messages, use the `verbose` parameter with the following values:

- 1  
Print all events while they are recorded/played back.
- 2  
Print also maintenance information of the internal ClockTimers. The default is disabled (0).

To disable instruction count checking (message 'instruction count differs'), set parameter `checkInstructionCount` to 0. The default is enabled.

## Integration

This component is intended to be a subcomponent of a visualisation component, for example a component that instantiates a Visualisation object using `createVisualisation()`. The integration is pretty light weight:

- Wire up the `control` and `ticks` ports.
- Use `control.registerVisRegion(regionPointer, regionName)` to register all relevant VisRegion pointers. You only need to register the regions that are used in the `processMessages()` function to identify a region by pointer.
- Call `control.putEvent()` for all `visEvents` as they come in regardless of where they come from, usually from `processMessages()`.
- Call `control.getEvent()` to retrieve recorded events (always called directly or indirectly by the callback (master) behavior `control.processEvents()`).

## Iris and MTI instances for VisEventRecorder

This model has the following Iris instances:

**Table 3-790: VisEventRecorder Iris instances**

InstanceName	ComponentName
VisEventRecorder	VisEventRecorder
VisEventRecorder.playbackDivider	ClockDivider
VisEventRecorder.playbackTimer	ClockTimerThread
VisEventRecorder.playbackTimer.timer	ClockTimerThread64
VisEventRecorder.playbackTimer.timer.thread	SchedulerThread
VisEventRecorder.playbackTimer.timer.thread_event	SchedulerThreadEvent
VisEventRecorder.recordingDivider	ClockDivider

This model has the following MTI trace components:

**Table 3-791: VisEventRecorder MTI instances**

InstanceName	ComponentName
VisEventRecorder.playbackDivider	ClockDivider
VisEventRecorder.recordingDivider	ClockDivider

### Ports for VisEventRecorder

**Table 3-792: Ports**

Name	Protocol	Type	Description
control	VisEventRecorderProtocol	Slave	The visualisation component controls the recorder through this port.
ticks	InstructionCount	Slave	Allow VisEventRecorder to get tick count from a core.

### Parameters for VisEventRecorder

#### checkInstructionCount

##### Type

bool

##### Default value

0x1

##### Description

check instruction count in recording file against actual instruction count during playback.

#### playbackDivider.div

##### Type

int

##### Default value

0x1

##### Description

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

#### playbackFileName

##### Type

string

##### Default value

""

##### Description

playback filename (empty string disables playback).

#### recordingDivider.div

##### Type

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADl and can only be set in LISA.

**recordingFileName****Type**

string

**Default value**

""

**Description**

recording filename (empty string disables recording).

**recordingTimeBase****Type**

int

**Default value**

0x5f5e100

**Description**

timebase in 1/s (relative to the master clock (e.g. 100000000 means 10 nanoseconds resolution simulated time for a 1Hz master clock)) to be used for recording (higher values -> higher time resolution, playback time base is always taken from the playback file).

**verbose****Type**

int

**Default value**

0x0

**Description**

enable verbose messages (1=normal, 2=even more).

### 3.7.76 Visualisation\_sdl2

Display window for VE using sdl2 Visualisation library. This model is written in C++.

**Iris and MTI instances for Visualisation\_sdl2**

This model has the following Iris instances:

**Table 3-793: Visualisation\_sdl2 Iris instances**

InstanceName	ComponentName
Visualisation_sdl2	Visualisation_sdl2
Visualisation_sdl2.recorder	VisEventRecorder
Visualisation_sdl2.recorder.playbackDivider	ClockDivider
Visualisation_sdl2.recorder.playbackTimer	ClockTimerThread
Visualisation_sdl2.recorder.playbackTimer.timer	ClockTimerThread64
Visualisation_sdl2.recorder.playbackTimer.timer.thread	SchedulerThread
Visualisation_sdl2.recorder.playbackTimer.timer.thread_event	SchedulerThreadEvent
Visualisation_sdl2.recorder.recordingDivider	ClockDivider

This model has the following MTI trace components:

**Table 3-794: Visualisation\_sdl2 MTI instances**

InstanceName	ComponentName
Visualisation_sdl2.recorder.playbackDivider	<a href="#">ClockDivider</a>
Visualisation_sdl2.recorder.recordingDivider	<a href="#">ClockDivider</a>

## Ports for Visualisation\_sdl2

**Table 3-795: Ports**

Name	Protocol	Type	Description
c0_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c10_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c11_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c12_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c13_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c14_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c15_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c1_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c2_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c3_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c4_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c5_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c6_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c7_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c8_core_freq[16]	<a href="#">ValueState</a>	Slave	-
c9_core_freq[16]	<a href="#">ValueState</a>	Slave	-
clock_50Hz	<a href="#">ClockSignal</a>	Slave	-
cluster0_ticks[16]	<a href="#">InstructionCount</a>	Slave	-
cluster10_ticks[16]	<a href="#">InstructionCount</a>	Slave	-
cluster11_ticks[16]	<a href="#">InstructionCount</a>	Slave	-

Name	Protocol	Type	Description
cluster12_ticks[16]	InstructionCount	Slave	-
cluster13_ticks[16]	InstructionCount	Slave	-
cluster14_ticks[16]	InstructionCount	Slave	-
cluster15_ticks[16]	InstructionCount	Slave	-
cluster1_ticks[16]	InstructionCount	Slave	-
cluster2_ticks[16]	InstructionCount	Slave	-
cluster3_ticks[16]	InstructionCount	Slave	-
cluster4_ticks[16]	InstructionCount	Slave	-
cluster5_ticks[16]	InstructionCount	Slave	-
cluster6_ticks[16]	InstructionCount	Slave	-
cluster7_ticks[16]	InstructionCount	Slave	-
cluster8_ticks[16]	InstructionCount	Slave	-
cluster9_ticks[16]	InstructionCount	Slave	-
cluster_freq[16]	ValueState	Slave	-
keyboard	KeyboardStatus	Master	-
lcd	LCD	Slave	-
lcd_layout	LCDLayoutInfo	Master	-
mcp_freq	ValueState	Slave	-
mcp_ticks	InstructionCount	Slave	-
mouse	MouseStatus	Master	-
poreset	Signal	Master	-
scp_freq	ValueState	Slave	-
scp_ticks	InstructionCount	Slave	-
sys_temperature[16]	ValueState	Slave	-
touch_screen	MouseStatus	Master	-

## Parameters for Visualisation\_sdl2

### **cluster0\_name**

#### Type

string

#### Default value

""

#### Description

Cluster0 name.

### **cluster10\_name**

#### Type

string

**Default value**

""

**Description**

Cluster10 name.

**cluster11\_name****Type**

string

**Default value**

""

**Description**

Cluster11 name.

**cluster12\_name****Type**

string

**Default value**

""

**Description**

Cluster12 name.

**cluster13\_name****Type**

string

**Default value**

""

**Description**

Cluster13 name.

**cluster14\_name****Type**

string

**Default value**

""

**Description**

Cluster14 name.

**cluster15\_name****Type**

string

**Default value**

""

**Description**

Cluster15 name.

**cluster1\_name****Type**

string

**Default value**

""

**Description**

Cluster1 name.

**cluster2\_name****Type**

string

**Default value**

""

**Description**

Cluster2 name.

**cluster3\_name****Type**

string

**Default value**

""

**Description**

Cluster3 name.

**cluster4\_name****Type**

string

**Default value**

""

**Description**

Cluster4 name.

**cluster5\_name****Type**

string

**Default value**

""

**Description**

Cluster5 name.

**cluster6\_name****Type**

string

**Default value**

""

**Description**

Cluster6 name.

**cluster7\_name****Type**

string

**Default value**

""

**Description**

Cluster7 name.

**cluster8\_name****Type**

string

**Default value**

""

**Description**

Cluster8 name.

**cluster9\_name****Type**

string

**Default value**

""

**Description**

Cluster9 name.

**css\_spec****Type**

string



**Default value**

"Columbus mid"

**Description**

Platform specification displayed in window title.

**diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

**disable\_visualisation****Type**

bool

**Default value**

0x0

**Description**

Enable/disable visualisation.

**display\_object****Type**

int

**Default value**

0x0

**Description**

Display objects: LCD only (1), status only (2), or both (0).

**display\_poreset\_button****Type**

bool

**Default value**

0x0

**Description**

Display power-on reset button.

**idler.delay\_ms****Type**

int

**Default value**

0x32

**Description**

GUI update period in ms.

**idler.has\_gui****Type**

bool

**Default value**

0x1

**Description**

GUI is enabled.

**is\_heterogeneous\_cluster****Type**

bool

**Default value**

0x0

**Description**

Is Heterogeneous cluster.

**lcd\_height\_param****Type**

int

**Default value**

0x258

**Description**

LCD Height.

**lcd\_width\_param****Type**

int

**Default value**

0x320

**Description**

LCD Width.

**mcp\_name****Type**

string

**Default value**  
"MCP: Cortex-M7"

**Description**  
MCP name.

**num\_cps**

**Type**  
int

**Default value**  
0x1

**Description**  
Number of Control Processors.

**per\_core\_clock**

**Type**  
bool

**Default value**  
0x0

**Description**  
Per-core clock connection.

**platform\_name**

**Type**  
string

**Default value**  
"Application Processors"

**Description**  
Platform Name.

**rate\_limit-enable**

**Type**  
bool

**Default value**  
0x0

**Description**  
Rate limit simulation.

**recorder.checkInstructionCount**

**Type**  
bool

**Default value**

0x1

**Description**

check instruction count in recording file against actual instruction count during playback.

**`recorder.playbackDivider.div`****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**`recorder.playbackDivider.mul`****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**`recorder.playbackFileName`****Type**

string

**Default value**

""

**Description**

playback filename (empty std::string disables playback).

**`recorder.recordingDivider.div`****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**`recorder.recordingDivider.mul`****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADL and can only be set in LISA.

**recorder.recordingFileName****Type**

string

**Default value**

""

**Description**

recording filename (empty std::string disables recording).

**recorder.recordingTimeBase****Type**

int

**Default value**

0x5f5e100

**Description**

timebase in 1/s (relative to the master clock (e.g. 100000000 means 10 nanoseconds resolution simulated time for a 1Hz master clock)) to be used for recording (higher values -> higher time resolution, playback time base is always taken from the playback file).

**recorder.verbose****Type**

int

**Default value**

0x0

**Description**

enable verbose messages (1=normal, 2=even more).

**scp\_name****Type**

string

**Default value**

"SCP: Cortex-M7"

**Description**

SCP name.

**shutdown\_pixel\_enable****Type**

bool

**Default value**

0x0

**Description**

Shutdown pixel enable. Use to trigger simulation shutdown when a specific pixel reaches the given target RGB value.

**shutdown\_pixel\_rgb****Type**

int

**Default value**

0xffffffff

**Description**

Shutdown pixel target RGB value 0xRRGGBB.

**shutdown\_pixel\_x****Type**

int

**Default value**

0x0

**Description**

Shutdown pixel X co-ordinate (0 is left).

**shutdown\_pixel\_y****Type**

int

**Default value**

0x0

**Description**

Shutdown pixel Y co-ordinate (0 is top).

**trap\_key****Type**

int

**Default value**

0x4a

**Description**  
Trap key that works with left Ctrl to toggle mouse display.

**window\_title**

**Type**  
string

**Default value**  
"Fast Models - %cpu%"

**Description**  
Window title(%cpu% will be replaced by css\_spec).

3.7.77 WarningMemory

Memory that prints warnings, and RAZ/WIs or aborts. This model is written in C++.

**Iris and MTI instances for WarningMemory**  
This model has the following Iris instances:

Table 3-796: WarningMemory Iris instances

InstanceName	ComponentName
WarningMemory	WarningMemory

Ports for WarningMemory

Table 3-797: Ports

Name	Protocol	Type	Description
pvbust	PVBus	Slave	Bus slave interface

Parameters for WarningMemory

**abort\_on\_reads**  
**Type**  
bool

**Default value**  
0x0

**Description**  
Generate Abort on reads.

**abort\_on\_writes**  
**Type**  
bool

**Default value**

0x0

**Description**

Generate Abort on writes.

**read\_data****Type**

int

**Default value**

0x0

**Description**

Data to return on reads, if not aborting.

**warn\_on\_reads****Type**

bool

**Default value**

0x1

**Description**

Generate Warn on reads.

**warn\_on\_writes****Type**

bool

**Default value**

0x1

**Description**

Generate Warn on writes.

**warning****Type**

string

**Default value**

"Invalid access"

**Description**

Warning string.



### 3.7.78 v8EmbeddedCrossTrigger\_Matrix

v8 Embedded Cross Trigger Matrix. This model is written in C++.

#### About v8EmbeddedCrossTrigger\_Matrix

This is a model of a platform-level Cross Trigger Matrix (CTM) for connection to the Cross Trigger Interface (CTI) ports provided on Armv8-A processors in Fast Models. The combination of the CTI and the CTM provides an architectural model of the Coresight embedded triggering system.

A single instance of the v8EmbeddedCrossTrigger\_Matrix component supports up to four clusters, each containing four cores. For example:

```
cluster0.cti[0] => v8ect.cti[0];
cluster0.cti[1] => v8ect.cti[1];
cluster0.cti[2] => v8ect.cti[2];
cluster0.cti[3] => v8ect.cti[3];
...
cluster3.cti[0] => v8ect.cti[12];
cluster3.cti[1] => v8ect.cti[13];
cluster3.cti[2] => v8ect.cti[14];
cluster3.cti[3] => v8ect.cti[15];
```

#### Iris and MTI instances for v8EmbeddedCrossTrigger\_Matrix

This model has the following Iris instances:

Table 3-798: v8EmbeddedCrossTrigger\_Matrix Iris instances

InstanceName	ComponentName
v8EmbeddedCrossTrigger_Matrix	EmbeddedCT

#### Ports for v8EmbeddedCrossTrigger\_Matrix

Table 3-799: Ports

Name	Protocol	Type	Description
cti[36]	v8EmbeddedCrossTrigger_controlprotocol	Slave	-

#### Parameters for v8EmbeddedCrossTrigger\_Matrix

##### has\_CTIAUTHSTATUS

###### Type

bool

###### Default value

0x1

###### Description

Has the optional CTIAUTHSTATUS register.

**has\_CTIDEVID\_INOUT****Type**

bool

**Default value**

0x1

**Description**

Has the option of input gate in cross trigger matrix.

**number-of-channels****Type**

int

**Default value**

0x4

**Description**

Number of channels in cross trigger matrix.

## 3.8 Scheduler components

This section describes the Scheduler components.

### 3.8.1 AsyncSignal

This model is written in C++.

#### About AsyncSignal

This component provides the means to cleanly schedule events from non-simulation threads onto the simulation thread.

#### Ports for AsyncSignal

**Table 3-800: Ports**

Name	Protocol	Type	Description
async_callback	<a href="#">AsyncSignalCallback</a>	Master	This port emits a call to <code>signal()</code> on the simulation thread asynchronously after <code>async_control.signal()</code> has been called.
async_control	<a href="#">AsyncSignalControl</a>	Slave	Non-simulation threads call <code>signal()</code> on this port in order to schedule an event: a call to <code>async_callback.signal()</code> on the simulation thread.

### 3.8.2 SchedulerInterface

A SchedulerInterface instance allows access to the Fast Models scheduler. This model is written in LISA+.

#### Iris and MTI instances for SchedulerInterface

This model has the following Iris instances:

**Table 3-801: SchedulerInterface Iris instances**

InstanceName	ComponentName
SchedulerInterface	SchedulerInterface

#### Ports for SchedulerInterface

**Table 3-802: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	Clock frequency for waitTicks() function.
control	<a href="#">SchedulerInterfaceControl</a>	Slave	Scheduler interface. Allows to: - wait for time

### 3.8.3 SchedulerThread

A SchedulerThread instance represents a co-routine thread in the simulation. This model is written in LISA+.

#### Iris and MTI instances for SchedulerThread

This model has the following Iris instances:

**Table 3-803: SchedulerThread Iris instances**

InstanceName	ComponentName
SchedulerThread	SchedulerThread

#### Ports for SchedulerThread

**Table 3-804: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	Clock frequency for waitTicks() function.
control	<a href="#">SchedulerThreadControl</a>	Slave	SchedulerThread control. Masters use this to: - control the thread (wait etc) - implement the actual thread function threadProc()

### 3.8.4 SchedulerThreadEvent

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on. This model is written in LISA+.

#### Iris and MTI instances for SchedulerThreadEvent

This model has the following Iris instances:

**Table 3-805: SchedulerThreadEvent Iris instances**

InstanceName	ComponentName
SchedulerThreadEvent	SchedulerThreadEvent

#### Ports for SchedulerThreadEvent

**Table 3-806: Ports**

Name	Protocol	Type	Description
control	<a href="#">SchedulerThreadEventControl</a>	Slave	SchedulerThreadEvent control. Masters use this to: - wait for this event - notify waiters that the event happened

## 3.9 Signals components

This section describes the Signals components.

### 3.9.1 AndGate

And Gate. This model is written in LISA+.

#### About AndGate

This component implements a logical AND of two signal input ports to generate a single output signal. For example, you can use it to combine two interrupt signals.

#### Iris and MTI instances for AndGate

This model has the following Iris instances:

**Table 3-807: AndGate Iris instances**

InstanceName	ComponentName
AndGate	AndGate

#### Ports for AndGate

**Table 3-808: Ports**

Name	Protocol	Type	Description
input[2]	<a href="#">Signal</a>	Slave	2 input signals to be AND'ed.
output	<a href="#">Signal</a>	Master	AND'ed output signal.

### 3.9.2 FrequencyProbe

Clock Frequency observer. This model is written in C++.

#### Iris and MTI instances for FrequencyProbe

This model has the following Iris instances:

**Table 3-809: FrequencyProbe Iris instances**

InstanceName	ComponentName
FrequencyProbe	FrequencyProbe

#### Ports for FrequencyProbe

**Table 3-810: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	-
freq_changed	<a href="#">ValueState</a>	Master	-

#### Parameters for FrequencyProbe

##### diagnostics

##### Type

int

##### Default value

0x0

##### Description

Diagnostics.

### 3.9.3 LabellerMasterIdExtendedIdUserFlag

Allows the modification of MasterID, ExtendedID and UserFlags attributes of PVBUS transactions. This model is written in LISA+.

#### Iris and MTI instances for LabellerMasterIdExtendedIdUserFlag

This model has the following Iris instances:

**Table 3-811: LabellerMasterIdExtendedIdUserFlag Iris instances**

InstanceName	ComponentName
LabellerMasterIdExtendedIdUserFlag	LabellerMasterIdExtendedIdUserFlag
LabellerMasterIdExtendedIdUserFlag.pvbuslogger	PVBusLogger
LabellerMasterIdExtendedIdUserFlag.pvbuslogger.mapper	PVBusMapper
LabellerMasterIdExtendedIdUserFlag.pvbusmodifier	PVBusMapper

This model has the following MTI trace components:

**Table 3-812: LabellerMasterIdExtendedIdUserFlag MTI instances**

InstanceName	ComponentName
LabellerMasterIdExtendedIdUserFlag.pvbuslogger	PVBusLogger
LabellerMasterIdExtendedIdUserFlag.pvbuslogger.mapper	PVBusMapper
LabellerMasterIdExtendedIdUserFlag.pvbusmodifier	PVBusMapper

### Ports for LabellerMasterIdExtendedIdUserFlag

**Table 3-813: Ports**

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Output with modified properties.
pvbus_s	PVBus	Slave	Unmodified input.

### Parameters for LabellerMasterIdExtendedIdUserFlag

#### **extendedid**

##### Type

int

##### Default value

0x0

##### Description

ExtendedID value to be applied to transactions.

#### **extendedid\_mask**

##### Type

int

##### Default value

0x0

##### Description

Mask used to determine which bits of extendedid parameter to be set in the transactions ExtendedID attribute. 0xFFFFFFFFFFFFFFFF will overwrite all the incoming ExtendedID bits with the value of the extendedid parameter, 0x0 will overwrite none.

#### **extendedmanagerid**

##### Type

int

##### Default value

0x0

##### Description

ExtendedManagerID value to be applied to transactions.

**extendedmanagerid\_mask****Type**

int

**Default value**

0x0

**Description**

Mask used to determine which bits of extendedmanagerid parameter to be set in the transactions ExtendedManagerID attribute. 0xFFFFFFFF will overwrite all the incoming ExtendedManagerID bits with the value of the extendedmanagerid parameter, 0x0 will overwrite none.

**masterid****Type**

int

**Default value**

0x0

**Description**

MasterID value to be applied to transactions.

**masterid\_mask****Type**

int

**Default value**

0x0

**Description**

Mask used to determine which bits of masterid parameter to be set in the transactions MasterID attribute. 0xFFFFFFFF will overwrite all the incoming MasterID bits with the value of the masterid parameter, 0x0 will overwrite none.

**pvbuslogger.trace\_debug****Type**

bool

**Default value**

0x0

**Description**

Enable tracing of debug transactions.

**pvbuslogger.trace\_snoops****Type**

bool

**Default value**

0x0

**Description**

Enable tracing of ACE snoop requests.

**userflags**

**Type**

int

**Default value**

0x0

**Description**

UserFlags value to be applied to transactions.

**userflags\_mask**

**Type**

int

**Default value**

0x0

**Description**

Mask used to determine which bits of userflags parameter to be set in the transactions UserFlags attribute. 0xFFFFFFFF will overwrite all the incoming UserFlags bits with the value of the userflags parameter, 0x0 will overwrite none.

3.9.4 LabellerUserSignals

This model is written in LISA+.

Iris and MTI instances for LabellerUserSignals

This model has the following Iris instances:

Table 3-814: LabellerUserSignals Iris instances

InstanceName	ComponentName
LabellerUserSignals	LabellerUserSignals
LabellerUserSignals.pvbuslogger	PVBusLogger
LabellerUserSignals.pvbuslogger.mapper	PVBusMapper
LabellerUserSignals.pvbusmodifier	PVBusMapper

This model has the following MTI trace components:



**Table 3-815: LabellerUserSignals MTI instances**

InstanceName	ComponentName
LabellerUserSignals.pvbuslogger	PVBusLogger
LabellerUserSignals.pvbuslogger.mapper	PVBusMapper
LabellerUserSignals.pvbusmodifier	PVBusMapper

### Ports for LabellerUserSignals

**Table 3-816: Ports**

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Output with modified UserFlags.
pvbus_s	PVBus	Slave	Unmodified input.

### Parameters for LabellerUserSignals

#### **pvbuslogger.trace\_debug**

##### Type

bool

##### Default value

0x0

##### Description

Enable tracing of debug transactions.

#### **pvbuslogger.trace\_snoops**

##### Type

bool

##### Default value

0x0

##### Description

Enable tracing of ACE snoop requests.

#### **user**

##### Type

int

##### Default value

0x0

##### Description

User signal to be applied to transactions.

### 3.9.5 OrGate

Or Gate. This model is written in LISA+.

#### About OrGate

This component implements a logical OR of two signal input ports to generate a single output signal. For example, you can use this component to combine two interrupt signals.

#### Iris and MTI instances for OrGate

This model has the following Iris instances:

**Table 3-817: OrGate Iris instances**

InstanceName	ComponentName
OrGate	OrGate

#### Ports for OrGate

**Table 3-818: Ports**

Name	Protocol	Type	Description
input[16]	Signal	Slave	16 input signals to be OR'ed.
output	Signal	Master	OR'ed output signal.

### 3.9.6 SGSignalBuffer

Buffer to synchronise SystemGenerator Signal setValue() calls. This model is written in LISA+.

#### About SGSignalBuffer

This component buffers changes to its input signal, outputting the new values at the next clock tick. This is useful in SystemC export situations as the output signal is driven from an SC\_THREAD whereas the input could have come from an SC\_METHOD. This avoids issues if a downstream component calls `sc_wait()`.



Note

There are variants of the buffer with different port array sizes.

#### Iris and MTI instances for SGSignalBuffer

This model has the following Iris instances:

**Table 3-819: SGSignalBuffer Iris instances**

InstanceName	ComponentName
SGSignalBuffer	SGSignalBuffer

## Ports for SGSignalBuffer

**Table 3-820: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	Clock rate to release buffered signals.
in	<a href="#">Signal</a>	Slave	Signal in.
out	<a href="#">Signal</a>	Master	Buffered signal out.

## 3.9.7 SignalDriver

Drives signal port based on parameter, register or bus slave port. This model is written in LISA+.

### Iris and MTI instances for SignalDriver

This model has the following Iris instances:

**Table 3-821: SignalDriver Iris instances**

InstanceName	ComponentName
SignalDriver	SignalDriver
SignalDriver.pvbuslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-822: SignalDriver MTI instances**

InstanceName	ComponentName
SignalDriver	<a href="#">SignalDriver</a>
SignalDriver.pvbuslave	<a href="#">PVBusSlave</a>

## Ports for SignalDriver

**Table 3-823: Ports**

Name	Protocol	Type	Description
pvbus_s	<a href="#">PVBus</a>	Slave	-
signal_out	<a href="#">Signal</a>	Master	-

## Parameters for SignalDriver

### param\_input

#### Type

bool

#### Default value

0x0

#### Description

Drive signal\_out port with this parameter value.

### 3.9.8 SignalInverter

This model is written in C++.

#### Iris and MTI instances for SignalInverter

This model has the following Iris instances:

**Table 3-824: SignalInverter Iris instances**

InstanceName	ComponentName
SignalInverter	SignalInverter

#### Ports for SignalInverter

**Table 3-825: Ports**

Name	Protocol	Type	Description
sig_in	Signal	Slave	-
sig_out	Signal	Master	-
sig_out_invert	Signal	Master	-

### 3.9.9 SignalLogger

Traces signal activity. This model is written in LISA+.

#### Iris and MTI instances for SignalLogger

This model has the following Iris instances:

**Table 3-826: SignalLogger Iris instances**

InstanceName	ComponentName
SignalLogger	SignalLogger

This model has the following MTI trace components:

**Table 3-827: SignalLogger MTI instances**

InstanceName	ComponentName
SignalLogger	SignalLogger

#### Ports for SignalLogger

**Table 3-828: Ports**

Name	Protocol	Type	Description
in	Signal	Slave	Input signal port.
out	Signal	Master	Output signal port.

## Parameters for SignalLogger

### **forward\_signal**

#### Type

bool

#### Default value

0x1

#### Description

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

## 3.9.10 Value64Logger

Traces value activity. This model is written in LISA+.

### Iris and MTI instances for Value64Logger

This model has the following Iris instances:

**Table 3-829: Value64Logger Iris instances**

InstanceName	ComponentName
Value64Logger	Value64Logger

This model has the following MTI trace components:

**Table 3-830: Value64Logger MTI instances**

InstanceName	ComponentName
Value64Logger	Value64Logger

## Ports for Value64Logger

**Table 3-831: Ports**

Name	Protocol	Type	Description
in	Value_64	Slave	Input signal port.
out	Value_64	Master	Output signal port.

## 3.9.11 ValueLogger

Traces value activity. This model is written in LISA+.

### Iris and MTI instances for ValueLogger

This model has the following Iris instances:

**Table 3-832: ValueLogger Iris instances**

InstanceName	ComponentName
ValueLogger	ValueLogger

This model has the following MTI trace components:

**Table 3-833: ValueLogger MTI instances**

InstanceName	ComponentName
ValueLogger	ValueLogger

### Ports for ValueLogger

**Table 3-834: Ports**

Name	Protocol	Type	Description
in	Value	Slave	Input signal port.
out	Value	Master	Output signal port.

## 3.9.12 WideAndGate

And Gate with up to 8 inputs. This model is written in C++.

### Iris and MTI instances for WideAndGate

This model has the following Iris instances:

**Table 3-835: WideAndGate Iris instances**

InstanceName	ComponentName
WideAndGate	WideAndGate

### Ports for WideAndGate

**Table 3-836: Ports**

Name	Protocol	Type	Description
input[8]	Signal	Slave	-
output	Signal	Master	-

### Parameters for WideAndGate

#### diagnostics

##### Type

int

##### Default value

0x0

**Description**

Diagnostics.

**3.9.13 WideOrGate**

Or Gate with up to 8 inputs. This model is written in C++.

**Iris and MTI instances for WideOrGate**

This model has the following Iris instances:

**Table 3-837: WideOrGate Iris instances**

InstanceName	ComponentName
WideOrGate	WideOrGate

**Ports for WideOrGate**

**Table 3-838: Ports**

Name	Protocol	Type	Description
input[8]	Signal	Slave	-
output	Signal	Master	-

**Parameters for WideOrGate**

**diagnostics**

**Type**

int

**Default value**

0x0

**Description**

Diagnostics.

**3.9.14 WideOrGate\_12x4**

Or Gate with up to 48 inputs and support to num\_cores. This model is written in C++.

**Iris and MTI instances for WideOrGate\_12x4**

This model has the following Iris instances:

**Table 3-839: WideOrGate\_12x4 Iris instances**

InstanceName	ComponentName
WideOrGate_12x4	WideOrGate_12x4

## Ports for WideOrGate\_12x4

Table 3-840: Ports

Name	Protocol	Type	Description
input[96]	Signal	Slave	-
output1	Signal	Master	-
output2	Signal	Master	-
output3	Signal	Master	-

## Parameters for WideOrGate\_12x4

**diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

**max\_cores\_per\_cluster****Type**

int

**Default value**

0x4

**Description**

Max number of cores per cluster.

**num\_cores****Type**

int

**Default value**

0x4

**Description**

Number of cores in cluster.

## 3.10 SystemIP components

This section describes the SystemIP components.

The major SystemIP components are:



- Input/output devices.
- Memory, including flash.
- Ethernet controller.
- Interrupt controllers.
- Static and dynamic memory controllers.
- Audio interface.
- Programmable clock generators.

These components are software implementations of specific hardware functionality.

### 3.10.1 AHCI\_SATA

AHCI controller with attached SATA disks and PCIe interface. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-841: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About AHCI\_SATA

This component implements an AHCI controller including attached SATA disks. It connects as a PCIe end-point device to a PCIe framework.

#### Iris and MTI instances for AHCI\_SATA

This model has the following Iris instances:

**Table 3-842: AHCI\_SATA Iris instances**

InstanceName	ComponentName
AHCI_SATA	AHCI_SATA
AHCI_SATA.ahci_master	PVBusMaster
AHCI_SATA.register_slave	PVBusSlave

This model has the following MTI trace components:

**Table 3-843: AHCI\_SATA MTI instances**

InstanceName	ComponentName
AHCI_SATA	<a href="#">ExportTestAHCI_SATA</a>
AHCI_SATA.ahci_master	<a href="#">PVBusMaster</a>
AHCI_SATA.register_slave	<a href="#">PVBusSlave</a>

## Ports for AHCI\_SATA

**Table 3-844: Ports**

Name	Protocol	Type	Description
ahci_dma_m	PVBus	Master	AHCI device performs DMA accesses via master
client_s	PCIDevice2ClientProtocol	Slave	PCIDevice client slave port, used for MSI-X
pvbus	PVBus	Slave	AHCI pci/control/config/status registers

## Parameters for AHCI\_SATA

### **force\_mode**

#### Type

string

#### Default value

"NCQ"

#### Description

Force disk to report support for at most PIO/DMA/NCQ mode (only for testing/bring-up purposes). PIO mode is always supported. Use NCQ for maximum performance (default).

### **image\_path**

#### Type

string

#### Default value

""

#### Description

Comma separated list of zero or more disk images (up to 32). Each image represents one SATA disk which is connected to one port of the AHCI controller. Empty list elements are allowed and result in a SATA port which has no disk attached. Empty string (default) means: One SATA port with no disk attached. Use 'truncate -s 4T disk.img' to create a 4 TByte sparse image. Use 'dd if=/dev/zero of=disk.img bs=1M count=42' to create a 42 MByte non-sparse image.

### **run\_async**

#### Type

bool

#### Default value

0x0

#### Description

Do host I/O in a background thread asynchronously. Enabling this makes the simulation non-deterministic and may or may not improve performance. Default is 'false' (do all disk accesses synchronously).

### 3.10.2 AddressTranslationUnit

Address Translator Unit. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-845: IP revisions support**

Revision	Quality level
0.5	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Changes in 11.29.19

Parameters added:

- `DEBUG_disable_translation`

#### Iris and MTI instances for AddressTranslationUnit

This model has the following Iris instances:

**Table 3-846: AddressTranslationUnit Iris instances**

InstanceName	ComponentName
AddressTranslationUnit	AddressTranslationUnit
AddressTranslationUnit.ATU_BusMapper	PVBusMapper
AddressTranslationUnit.apb	PVBusSlave

This model has the following MTI trace components:

**Table 3-847: AddressTranslationUnit MTI instances**

InstanceName	ComponentName
AddressTranslationUnit.ATU_BusMapper	<a href="#">PVBusMapper</a>
AddressTranslationUnit.apb	<a href="#">PVBusSlave</a>

#### Ports for AddressTranslationUnit

**Table 3-848: Ports**

Name	Protocol	Type	Description
apb	<a href="#">PVBus</a>	Slave	-
irq_out	<a href="#">Signal</a>	Master	-
pdbus_m	<a href="#">PVBus</a>	Master	-
pdbus_s	<a href="#">PVBus</a>	Slave	-
reset_in	<a href="#">Signal</a>	Slave	-

## Parameters for AddressTranslationUnit

### **ATUNTR**

#### Type

int

#### Default value

0x5

#### Description

Number of translation regions (1=2, 2=4, 3=8, 4=16, and 5=32).

### **ATUPAW**

#### Type

int

#### Default value

0x5

#### Description

Physical address width (0=32, 1=36, 2=40, 3=44, 4=48, 5=52, 6=56, 7=64 bits). No impact on the PVBUS transactions.

### **ATUPS**

#### Type

int

#### Default value

0xd

#### Description

Selects the page size granularity in bytes (0xC=4096, 0xD=8192, and 0xE=16384) (Default=0xD).

### **DEBUG\_disable\_translation**

#### Type

bool

#### Default value

0x0

#### Description

A model-only parameter to disable any remapping by the ATU, despite register configuration. For debug/test only.

### **diagnostics**

#### Type

int

**Default value**

0x0

**Description**

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

### 3.10.3 BP141\_TZMA

PrimeCell Infrastructure AMBA 3 AXI TrustZone Memory Adapter. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-849: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About BP141\_TZMA**

BP141\_TZMA permits a single physical memory cell of up to 2 MB to be shared between a secure and non-secure storage area. The partitioning between these areas is flexible.

This component routes transactions according to the following:

- The memory region that they are attempting to access.
- The security mode of the transaction.

The BP141\_TZMA fixes the base address of the secure region to the base address of the decode space. It uses the R0SIZE [9:0] input to configure the size of the secure region in 4 KB increments up to a maximum of 2 MB.

TZMEMSIZE is the maximum addressing range of the memory as defined by that parameter. By default, TZMEMSIZE is set to 2 MB. In the following table, AxADDR is the offset address that the transactions want to access:

**Table 3-850: BP141\_TZMA security control**

AxADDR	Memory Region	Non-secure Transfer	Secure Transfer
AxADDR < R0Size	Secure, R0	Illegal	Legal
R0SIZE <= AxADDR and AxADDR < TZMEMSIZE	Non-secure, R1	Legal	Legal
AxADDR => TZMEMSIZE	No access	Illegal	Illegal

**Iris and MTI instances for BP141\_TZMA**

This model has the following Iris instances:

**Table 3-851: BP141\_TZMA Iris instances**

InstanceName	ComponentName
BP141_TZMA	BP141_TZMA
BP141_TZMA.pvbusrange_0	PVBusRange
BP141_TZMA.pvbusrange_0.pvbus_mapper	PVBusMapper
BP141_TZMA.tzswitch_0	TZSwitch
BP141_TZMA.tzswitch_0.pvbus_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-852: BP141\_TZMA MTI instances**

InstanceName	ComponentName
BP141_TZMA.pvbusrange_0.pvbus_mapper	PVBusMapper
BP141_TZMA.tzswitch_0.pvbus_mapper	PVBusMapper

## Ports for BP141\_TZMA

**Table 3-853: Ports**

Name	Protocol	Type	Description
pv_output	PVBus	Master	Routed PVBus output
pvbus	PVBus	Slave	Bus slave interface.
R0Size	Value	Slave	A software interface that is driven from the TrustZone Protection Controller (TZPC), setting the secure region size by bits[9:0].

## Parameters for BP141\_TZMA

### TZMEMSIZE

#### Type

int

#### Default value

0x200000

#### Description

Addressable range of device.

### TZSECROMSIZE

#### Type

int

#### Default value

0x200

#### Description

Default secure size.

**TZSEGSIZE**

**Type**  
int

**Default value**  
0x1000

**Description**  
Segment size.

**pvbusrange\_0.range**

**Type**  
int

**Default value**  
0x0

**Description**  
Addressable range routed to pvbus\_port\_a.

3.10.4 BP147\_TZPC

TrustZone Protection Controller. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-854: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About BP147\_TZPC

BP147\_TZPC provides a software interface to the protection bits in a secure system in a TrustZone design.

Iris and MTI instances for BP147\_TZPC

This model has the following Iris instances:

Table 3-855: BP147\_TZPC Iris instances

InstanceName	ComponentName
BP147_TZPC	BP147_TZPC
BP147_TZPC.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-856: BP147\_TZPC MTI instances**

InstanceName	ComponentName
BP147_TZPC.busslave	PVBusSlave

### Ports for BP147\_TZPC

**Table 3-857: Ports**

Name	Protocol	Type	Description
bus_in_s	PVBus	Slave	Slave port for register access.
TZPCDECPROT0	Value	Master	Output decode protection 0 status.
TZPCDECPROT1	Value	Master	Output decode protection 1 status.
TZPCDECPROT2	Value	Master	Output decode protection 2 status.
TZPCR0SIZE	Value	Master	Output secure RAM region size.

## 3.10.5 CCI400

Cache Coherent Interconnect for AXI4 ACE. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-858: IP revisions support**

Revision	Quality level
rOp0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About CCI400

If you disable the `cache_state_modelled` parameter, this component has negligible performance impact. If you enable `cache_state_modelled`, it adds significant cost to throughput for coherent transactions.

This model implements the slave interface Shareable Override Register, which can be read and written, but it has no functionality.

### ACE limitation

AXI Coherency Extensions (ACE) are extensions to AXI4 that support system-level cache coherency between multiple clusters. The ACE cache models in the Arm Cortex-A15 and Cortex-A7, and the ACE support in the CCI-400 have the limitation that they only process one transaction at a time. Normally, the simulation processes each transaction to completion before allowing any master to generate another transaction.

However, in the following situation the simulation might fail. If a SystemC bus slave calls `wait()` while it is processing a transaction, this call might allow another master to issue another transaction that passes through the CCI-400 or the Cortex-A15 or Cortex-A7 caches. This situation could



happen if a SystemC bus master running in another thread is connected to one of the ACE-lite ports on the CCI-400.

### Iris and MTI instances for CCI400

This model has the following Iris instances:

**Table 3-859: CCI400 Iris instances**

InstanceName	ComponentName
CCI400	CCI400
CCI400.cciinterconnect	PVCache
CCI400.cciregisters	CCIRegisters
CCI400.cciregisters.clocktimer	ClockTimerThread
CCI400.cciregisters.clocktimer.timer	ClockTimerThread64
CCI400.cciregisters.clocktimer.timer.thread	SchedulerThread
CCI400.cciregisters.clocktimer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-860: CCI400 MTI instances**

InstanceName	ComponentName
CCI400	CCI400

### Ports for CCI400

**Table 3-861: Ports**

Name	Protocol	Type	Description
acchannelen	Value	Slave	For each upstream port, determine if it is enabled or not with respect to snoop requests.
barrierterminate	Value	Slave	For each downstream port, determine if barriers are terminated at that port.
broadcastcachemain	Value	Slave	For each downstream port, determine if broadcast cache maintenance operations are forwarded down that port. A three bit signal but as the model only have a single downstream port, setting any of the bits will make it work.
bufferableoverride	Value	Slave	For each downstream port, determine if all transactions are forced to non-bufferable (AWCACHE[0] is forced to 0).
clk_in	ClockSignal	Slave	Clock signal for cciregisters
errorirq	Signal	Master	A signal stating that the imprecise error register is nonzero.
evntcntoverflow[5]	Signal	Master	When an event counter overflows, it sets the corresponding signal.
lint_ace_3_reset_state	Signal	Slave	This port can be connected to the reset signals of the system attached to the pvbus_s_ace_3 port.
lint_ace_4_reset_state	Signal	Slave	This port can be connected to the reset signals of the system attached to the pvbus_s_ace_4 port.
periphbase	Value_64	Slave	This port sets the base address of the private peripheral region.
pvbus_m	PVBus	Master	Master port for all downstream memory accesses.
pvbus_s_ace_3	PVBus	Slave	ACE-capable slave ports.

Name	Protocol	Type	Description
pvbus_s_ace_4	PVBus	Slave	ACE-capable slave ports.
pvbus_s_ace_lite_plus_dvm_0	PVBus	Slave	Memory bus interface that implements ACE lite and DVM protocol.
pvbus_s_ace_lite_plus_dvm_1	PVBus	Slave	Memory bus interface that implements ACE lite and DVM protocol.
pvbus_s_ace_lite_plus_dvm_2	PVBus	Slave	Memory bus interface that implements ACE lite and DVM protocol.
reset_in	Signal	Slave	Signal to reset the CCI.
reset_state_of_ace_lite_ports[3]	Signal	Slave	This port can be connected to the reset signals of the system attached to ACE-Lite ports 0,1,2

## Parameters for CCI400

### **acchannelen**

#### Type

int

#### Default value

0x1f

#### Description

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

### **barrierterminate**

#### Type

int

#### Default value

0x7

#### Description

For each downstream port, determine if barriers will be terminated at that port.

### **broadcastcachemain**

#### Type

int

#### Default value

0x0

#### Description

For each downstream port a bit determines if broadcast cache maintenance operations are forwarded down that port.

### **bufferableoverride**

#### Type

int

**Default value**

0x0

**Description**

For each downstream port, determine if all transactions will be forced to non-bufferable.

**cache\_state\_modelled****Type**

bool

**Default value**

0x1

**Description**

Model the cache state.

**force\_on\_from\_start****Type**

bool

**Default value**

false

Enables snooping on upstream ports from the start of simulation.

The CCI will normally start up with snooping disabled, however, using this parameter we allow the model to start with snooping enabled without SW drivers programming the CCI.

This is only setup at simulation reset and not at signal reset.

If the upstreams can ever be held in reset then you *must* connect:

- `reset_state_of_ace_lite_ports[]`
- `lint_ace_3_reset_state`
- `lint_ace_4_reset_state`

so that it knows when to disable snoops to the upstream systems.

Otherwise, the upstream system will receive snoop messages whilst in reset and complain that it 'received a snoop request whilst it was in reset'.

**is\_downstream\_domain\_boundary\_for\_far\_atomic****Type**

bool

**Default value**

0x0

**Description**

This interconnect is at the last stage of the domain boundary.

**log\_enabled****Type**

int

**Default value**

1

Enable log messages from the CCI register file.

**0**

do not print anything

**1**

print only access violations

**2**

also print writes

**3**

print reads as well.

**periphbase****Type**

int

**Default value**

0x2c000000

**Description**

Value for PERIPHBASE. Only bits [39:16] are used. This value may be overridden by an input on the periphbase port.

**revision****Type**

string

**Default value**

"rOp0"

**Description**

Revision of the CCI400.

### 3.10.6 CCI500

Cache Coherent Interconnect. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-862: IP revisions support**

Revision	Quality level
r0p0	Full support
r0p2	Full support
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About CCI500

The LISA file declares seven upstream ports. You can configure these ports with `num_ace_ports` and `num_ace_lite_ports`:

- The bottom `num_ace_lite_ports` are ACE-Lite+DVM.
- The next `num_ace_ports` are ACE.
- Any remaining ports are ignored. If transactions are made on them, then warnings are produced.

For example, if `num_ace_ports = 1` and `num_ace_lite_ports = 1` then

- `pvbus_s[1]` is ACE
- `pvbus_s[0]` is ACE-Lite+DVM
- `pvbus_s[6-2]` are considered not to exist.

#### Differences between the model and the RTL

##### Address Decoder

- Only supports striping down to 4KiB.
- If the address decoder aborts the access, returns `SLVERR` rather than `DECERR`.

##### Interfaces

- The model does not implement the Q-Channel and P-Channel interfaces.

##### Performance Monitoring Unit

- PMU counters recognize only a few event sources:
  - Slave interface events:

**3**

ReadOnce.

**4**

ReadClean, ReadShared, ReadNotSharedDirty, ReadUnique.

5

MakeUnique, CleanUnique.

6

CleanInvalid, CleanShared, MakeInvalid.

7

DVM transaction received from upstream.

9

Read data that is satisfied by a snoop request.

- No events are implemented for the global events or for the master events.
- The PMU does not implement the event bus (EVNTBUS).

### Register access

- The register file only supports 32-bit accesses to its registers. Later versions of the CCI500 hardware support full write strobes to the register file. This limitation means that byte and halfword accesses work on the hardware but not on this version of the component.

### Snoop filter RAMs

- Snoop filter RAMs are not modeled. The Status Register fields that relate to the power state of the Snoop filter RAMs are undefined.

### Iris and MTI instances for CCI500

This model has the following Iris instances:

**Table 3-863: CCI500 Iris instances**

InstanceName	ComponentName
CCI500	CCI500
CCI500.pvbus_register_file_s[0]	PVBusSlave
CCI500.upstream[0]	PVBusSlave
CCI500.upstream[1]	PVBusSlave
CCI500.upstream[2]	PVBusSlave
CCI500.upstream[3]	PVBusSlave
CCI500.upstream[4]	PVBusSlave
CCI500.upstream[5]	PVBusSlave
CCI500.upstream[6]	PVBusSlave

This model has the following MTI trace components:

**Table 3-864: CCI500 MTI instances**

InstanceName	ComponentName
CCI500	CCI500
CCI500.pvbus_register_file_s[0]	PVBusSlave
CCI500.upstream[0]	PVBusSlave

InstanceName	ComponentName
CCI500.upstream[1]	PVBusSlave
CCI500.upstream[2]	PVBusSlave
CCI500.upstream[3]	PVBusSlave
CCI500.upstream[4]	PVBusSlave
CCI500.upstream[5]	PVBusSlave
CCI500.upstream[6]	PVBusSlave

## Ports for CCI500

**Table 3-865: Ports**

Name	Protocol	Type	Description
acchannelensx[7]	Value	Slave	ACCHANNELENSx represents the ports ACCHANNELENS0..ACCHANNELENS7 on the RTL (assuming there are seven upstream ports). * each upstream ACE port 'y' (pvbus_s[y]) has a two bit ACCHANNELENSx * bit 0 == 0 -- DVM messages are disabled from being sent to this interface * bit 1 == 0 -- Snoop messages are disabled from being sent to this interface * each upstream ACE-Lite port 'z' (pvbus_s[z]) has a one bit ACCHANNELENSx * bit 0 == 0 -- DVM messages are disabled from being sent to this interface In the model, as we support a variety of configurations with a single LISA file then each port will behave as though it is one bit or two bit as appropriate. If you send a value that cannot be represented, given the width of the port, then the CCI model will halt and produce a fatal error. The assumed values of these are set by parameters until they are driven, so you need not drive them if they are constant. In the RTL, these signals are sampled at reset. Due to ordering issues w.r.t. reset() on different components then we cannot do that. Instead the signals are sampled at first transaction. Thus any controller that is producing these signals has to hold them constant for long enough. AC channel enables.
address_decoder	CCI500_AddressDecoderProtocol	Master	An address decoder can be attached to the address_decoder port to choose which pvbus_s port a downstream transaction will go out of. If you do not connect an address decoder then all transactions will go out of port 0.
dbgen	Signal	Slave	Invasive debug enable.
errirq	Signal	Master	Indicates that an error response, DECERR or SLVERR, is received on the RRESP, BRESP, or CRRESP input signals, and it cannot be signaled precisely.
evntcntoverflow[8]	Signal	Master	Overflow flags for the PMU clock and counters.
niden	Signal	Slave	Non-invasive debug enable.
pvbus_m[6]	PVBus	Master	Bus master ports.
pvbus_register_file_s	PVBus	Slave	The slave port of the register file.
pvbus_s[7]	PVBus	Slave	Bus slave ports.
reset_in	Signal	Slave	Reset the interconnect.

Name	Protocol	Type	Description
reset_state_of_upstream_port[7]	Signal	Slave	Tell the interconnect the reset state of the upstream ports, this can be used by the interconnect to check some aspects of the reset sequencing. If you are using force_on_from_start then you <u>must</u> connect these pins.
spiden	Signal	Slave	Secure invasive debug enable.
spniden	Signal	Slave	Secure privileged non-invasive debug enable.

## Parameters for CCI500

### acchannelens0

#### Type

uint64\_t

#### Default value

0x0

For upstream port `pvtbus_s[0]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[0]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of 0x3 create a functional system without excessive configuration.

### acchannelens1

#### Type

uint64\_t

#### Default value

0x0

For upstream port `pvtbus_s[1]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[1]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of 0x3 create a functional system without excessive configuration.



**acchannelens2****Type**

uint64\_t

**Default value**

0x0

For upstream port `pvbuss[2]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[2]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

**acchannelens3****Type**

uint64\_t

**Default value**

0x0

For upstream port `pvbuss[3]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[3]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

**acchannelens4****Type**

uint64\_t

**Default value**

0x0

For upstream port `pvbuss[4]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[4]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

### **acchannelens5**

#### **Type**

`uint64_t`

#### **Default value**

`0x0`

For upstream port `pvtbus_s[5]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[5]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

### **acchannelens6**

#### **Type**

`uint64_t`

#### **Default value**

`0x0`

For upstream port `pvtbus_s[6]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[6]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

### **addr\_width**

#### **Type**

`int`

#### **Default value**

`0x28`

**Description**

The bit-width of the address that the CCI can accept.

**cache\_state\_modelled****Type**

bool

**Default value**

0x1

**Description**

Model the cache state.

**dbgen****Type**

bool

**Default value**

0x1

**Description**

Invasive debug enable. If true, enables the counting of PMU events.

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enable PVBUSLoggers for the downstream ports in the CCI model.

**force\_on\_from\_start****Type**

bool

**Default value**

false

Enables snooping on upstream ports from the start of simulation.

The interconnect will normally start up with snooping/DVM disabled. This parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if ACCHANNELENSx allows it.

No software driver for the interconnect is needed.

Any port that could go into reset must have `reset_state_of_upstream_port[]` reflect the reset state of that upstream system.

Otherwise, the upstream system may receive snoop/DVM messages whilst in reset and may complain that it 'received a snoop request whilst it was in reset'.

Do not use if software is directly controlling the interconnect.

This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.

### **niden**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

Whether non-secure events are allowed to be counted in the performance monitor.

### **num\_ace\_lite\_ports**

#### **Type**

int

#### **Default value**

0x5

#### **Description**

The bottom `num_ace_lite_ports` are ACE-Lite+DVM.

### **num\_ace\_ports**

#### **Type**

int

#### **Default value**

0x2

#### **Description**

The top `num_ace_ports` are ACE and support full coherency.

### **number\_of\_phantom\_entries**

#### **Type**

uint64\_t

#### **Default value**

0x0000000000000020

“Number of phantom entries in the cache. Phantom entries are used by certain cache operations to hold temporary data. Usually this should be left at the default value which is safe for all systems containing up to 32 masters.

**qos\_threshold\_upper****Type**

int

**Default value**

0xc

**Description**

Reset value for the QoS threshold register.

**reentrancy\_support****Type**

string

**Default value**

“env”

Must be one of:

**on**

hazard checking per cache line (normal mode)

**off**

no hazard checking (use only for single master systems)

**cacheglobal**

hazard checking globally for cache (not per cache line, testing feature, provokes more hazards than necessary)

**env**take value from `FM_REENTRANCY_SUPPORT` env var, if this is not set use ‘on’**spiden****Type**

bool

**Default value**

true

Secure invasive debug enable. If both `SPIDEN` and `DBGEN` are high, enables the counting of both Non-secure and Secure events.

**spniden****Type**

bool

**Default value**

true

Whether secure and non-secure events are allowed to be counted in the performance monitor

**version**

**Type**

string

**Default value**

«»

The version of the interconnect.

Allowed versions are:

- r0p0
- r0p2
- r1p0

3.10.7 CCI550

Cache Coherent Interconnect for AXI4 ACE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-866: IP revisions support

Revision	Quality level
r0p0	Full support
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Differences between the model and the RTL

Address Decoder

- Only supports striping down to 4KiB.
- If the address decoder aborts the access, returns SLVERR rather than DECERR.

Interfaces

- The model does not implement the Q-Channel and P-Channel interfaces.

Performance Monitoring Unit

- PMU counters recognize only a few event sources:
  - Slave interface events:

- 3      ReadOnce.
- 4      ReadClean, ReadShared, ReadNotSharedDirty, ReadUnique.
- 5      MakeUnique, CleanUnique.
- 6      CleanInvalid, CleanShared, MakeInvalid.
- 7      DVM transaction received from upstream.
- 9      Read data that is satisfied by a snoop request.

- No events are implemented for the global events or for the master events.
- The PMU does not implement the event bus (EVNTBUS).

### Reset signal sampling

- The configuration ports `acchannelensx[]` are sampled in the hardware when coming out of reset. In the model, these ports are sampled at the first transaction to a `pvbus_s` port or to the register file.

### Status Register, change-pending, and DVM messages

- The Status Register provides information on when the last transaction that could have observed an old value of a snoop or DVM enable has finished in the upstream system. Therefore a port that has been disabled can now have the system upstream of that port turned off. The model does not track DVM messages in the upstream system.

### Snoop filter RAMs

- The CCI-550 hardware has a snoop filter that reduces the number of snoop requests that the interconnect has to make. The model does not have a snoop filter and could make more snoop requests than the hardware would. This difference has no programmer-visible effect.
- The Status Register fields that relate to the power state of the Snoop filter RAMs are undefined.

### Registers

- The following registers provide storage but have no effect on the model.
  - QoS registers.
  - Interface monitor registers. These registers are intended for silicon debug.

### Iris and MTI instances for CCI550

This model has the following Iris instances:

**Table 3-867: CCI550 Iris instances**

InstanceName	ComponentName
CCI550	CCI550
CCI550.pvbus_register_file_s[0]	PVBusSlave
CCI550.upstream[0]	PVBusSlave
CCI550.upstream[1]	PVBusSlave
CCI550.upstream[2]	PVBusSlave
CCI550.upstream[3]	PVBusSlave
CCI550.upstream[4]	PVBusSlave
CCI550.upstream[5]	PVBusSlave
CCI550.upstream[6]	PVBusSlave

This model has the following MTI trace components:

**Table 3-868: CCI550 MTI instances**

InstanceName	ComponentName
CCI550	CCI550
CCI550.pvbus_register_file_s[0]	PVBusSlave
CCI550.upstream[0]	PVBusSlave
CCI550.upstream[1]	PVBusSlave
CCI550.upstream[2]	PVBusSlave
CCI550.upstream[3]	PVBusSlave
CCI550.upstream[4]	PVBusSlave
CCI550.upstream[5]	PVBusSlave
CCI550.upstream[6]	PVBusSlave

## Ports for CCI550

**Table 3-869: Ports**

Name	Protocol	Type	Description
acchannelensx[7]	Value	Slave	The acchannelensx[N] pins are used to tell the interconnect if the upstream system will accept snoops and/or DVM messages.
address_decoder	CCI500_AddressDecoderProtocol	Master	An address decoder can be attached to the address_decoder port to choose which pvbus_s port a downstream transaction will go out of. If you do not connect an address decoder then all transactions will go out of port 0.
dbgen	Signal	Slave	Invasive debug enable.
errirq	Signal	Master	Some async error was detected.
evntcntoverflow[8]	Signal	Master	The output interrupts of the event counters.
niden	Signal	Slave	Non-invasive debug enable.
pvbus_m[7]	PVBus	Master	The downstream master ports.



Name	Protocol	Type	Description
pvbus_register_file_s	PVBus	Slave	The slave port of the register file.
pvbus_s[7]	PVBus	Slave	Bus slave ports.
reset_in	Signal	Slave	Reset the interconnect.
reset_state_of_upstream_port[7]	Signal	Slave	Tell the interconnect the reset state of the upstream ports, this can be used by the interconnect to check some aspects of the reset sequencing. If you are using force_on_from_start then you _must_ connect these pins.
sci_s[7]	SystemCoherencyInterface	Slave	The System Coherency Interface bus. For those upstream ports that have a corresponding bit set in the bitmap of si_system_coherency_interface then the corresponding sci_m port can be used to move the upstream system into and out of the coherency domain.
spiden	Signal	Slave	Secure privileged invasive debug enable.
spniden	Signal	Slave	Secure privileged non-invasive debug enable.

## Parameters for CCI550

### acchannelens0

#### Type

int

#### Default value

0x0

#### Description

For upstream port 0 determine if it is enabled or not w.r.t. snoop requests.

### acchannelens1

#### Type

int

#### Default value

0x0

#### Description

For upstream port 1 determine if it is enabled or not w.r.t. snoop requests.

### acchannelens2

#### Type

int

#### Default value

0x0

#### Description

For upstream port 2 determine if it is enabled or not w.r.t. snoop requests.

**acchannelens3****Type**

int

**Default value**

0x0

**Description**

For upstream port 3 determine if it is enabled or not w.r.t. snoop requests.

**acchannelens4****Type**

int

**Default value**

0x0

**Description**

For upstream port 4 determine if it is enabled or not w.r.t. snoop requests.

**acchannelens5****Type**

int

**Default value**

0x0

**Description**

For upstream port 5 determine if it is enabled or not w.r.t. snoop requests.

**acchannelens6****Type**

int

**Default value**

0x0

**Description**

For upstream port 6 determine if it is enabled or not w.r.t. snoop requests.

**addr\_width****Type**

int

**Default value**

0x28

**Description**

The bit-width of the address that the CCI can accept.

**cache\_state\_modelled****Type**

bool

**Default value**

0x1

**Description**

Model the cache state.

**dbgen****Type**

bool

**Default value**

0x1

**Description**

Invasive debug enable. If true, enables the counting of PMU events.

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enable PVBUSLoggers for the downstream ports in the CCI model.

**force\_on\_from\_start****Type**

bool

**Default value**

false

The interconnect will normally start up with snooping/DVM disabled.

The parameter `si_system_coherency_interface` determines which connections are managed by the System Coherency Interface (SCI).

For connections that are managed by SCI, then this parameter has no effect.

For all other connections, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `ACCHANNELSENSx` allows it.

No software driver for the interconnect is needed.

Any non-SCI port that could go into reset must have `reset_state_of_upstream_port[]` reflect the reset state of that upstream.

Otherwise, the upstream system may receive snoop/DVM messages whilst in reset and may complain that it 'received a snoop request whilst it was in reset'.

Do not use if software is directly controlling the interconnect. This option does not disavow responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.

### **niden**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

Whether non-secure events are allowed to be counted in the performance monitor.

### **num\_ace\_lite\_ports**

#### **Type**

uint64\_t

#### **Default value**

0x0000000000000005

The bottom `num_ace_lite_ports` are ACE-Lite+DVM.

### **num\_ace\_ports**

#### **Type**

uint64\_t

#### **Default value**

0x0000000000000002

The top `num_ace_ports` are ACE and support full coherency.

### **number\_of\_phantom\_entries**

#### **Type**

uint64\_t

#### **Default value**

0x0000000000000020

Number of phantom entries in the cache. Phantom entries are used by certain cache operations to hold temporary data. Usually this should be left at the default value which is safe for all systems containing up to 32 masters.

**qos\_threshold\_upper****Type**

int

**Default value**

0xc

**Description**

Reset value for the QoS threshold register.

**reentrancy\_support****Type**

string

**Default value**

"env"

Must be one of:

**on**

hazard checking per cache line (normal mode)

**off**

no hazard checking (use only for single master systems)

**cacheglobal**

hazard checking globally for cache (not per cache line, testing feature, provokes more hazards than necessary)

**env**take value from `FM_REENTRANCY_SUPPORT` env var, if this is not set use 'on'**si0\_qos\_bw\_regulator****Type**

bool

**Default value**

0x0

**Description**

For upstream port 0 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

**si1\_qos\_bw\_regulator****Type**

bool

**Default value**

0x0

**Description**

For upstream port 1 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

**si2\_qos\_bw\_regulator****Type**

bool

**Default value**

0x0

**Description**

For upstream port 2 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

**si3\_qos\_bw\_regulator****Type**

bool

**Default value**

0x0

**Description**

For upstream port 3 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

**si4\_qos\_bw\_regulator****Type**

bool

**Default value**

0x0

**Description**

For upstream port 4 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

**si5\_qos\_bw\_regulator****Type**

bool

**Default value**

0x0

**Description**

For upstream port 5 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

**si6\_qos\_bw\_regulator****Type**

bool

**Default value**

0x0

**Description**

For upstream port 6 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

**si\_system\_coherency\_interface****Type**

uint64\_t

**Default value**

0x0000000000000000

This parameter tells the interconnect which upstream ports should be controlled by the System Coherency Interface.

Each bit corresponds to an upstream port, bit 0 to upstream port 0, etc.

If the SCI port is connected but `si_system_coherency_interface` disables its use then messages from the upstream will be ignored and software must manage the upstream system's entrance and exit of the coherency domain.

**spiden****Type**

bool

**Default value**

true

Secure invasive debug enable.

If both `SPIDEN` and `DBGEN` are high, enables the counting of both Non-secure and Secure events.

**spniden****Type**

bool

**Default value**

true

Whether secure and non-secure events are allowed to be counted in the performance monitor

**version**

**Type**  
string

**Default value**  
"0"

The version of the interconnect. Allowed versions are:

- r0p0
- r1p0

3.10.8 CCN502

CCN502 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-870: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About CCN502

CCN502 can be used for connecting components using the ACE and ACE-Lite interfaces. It has an L3 cache that can provide coherency between up to four fully-coherent ACE clusters and nine I/O coherent masters. It can connect up to two memory elements to drive transaction requests. This interconnect can be configured to support six or eight crosspoints, both of which are implemented in the model.

CCN502 has three or five downstream ports, depending on the number of crosspoints:

- Two or four SN-F ports for the memory controller
- One Acelite port (HNI)

The CCN502 parameters are not exposed through CADI, but can be seen in `ccn502.lisa`. Some useful parameters are:

- `cache_state_modelled`
- `cache_size_in_kbytes`
- `systemaddrmap`
- `variant_name`

CCN-502 supports up to 4 SN-Fs. To enable 4 SN-Fs in the model, set the `variant_name` parameter to `CCN502_8XP`, which means the 8XP/4HNF configuration.



## Limitations

The model has the following limitations:

- No support for 3 SN stripping.
- If there are multiple SN-Fs, the distribution of addresses to each SN-F is not guaranteed to match the hardware.
- The parameters for the CCNCache subcomponent are not accessible in System Canvas.

## Iris and MTI instances for CCN502

This model has the following Iris instances:

**Table 3-871: CCN502 Iris instances**

InstanceName	ComponentName
CCN502	CCN5XX
CCN502.bus_slave_ocm	PVBusSlave
CCN502.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN502.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN502.ccn_cache	CCNCache
CCN502.ccn_cache.upstream[0]	PVBusSlave
CCN502.ccn_cache.upstream[10]	PVBusSlave
CCN502.ccn_cache.upstream[11]	PVBusSlave
CCN502.ccn_cache.upstream[12]	PVBusSlave
CCN502.ccn_cache.upstream[13]	PVBusSlave
CCN502.ccn_cache.upstream[14]	PVBusSlave
CCN502.ccn_cache.upstream[15]	PVBusSlave
CCN502.ccn_cache.upstream[16]	PVBusSlave
CCN502.ccn_cache.upstream[17]	PVBusSlave
CCN502.ccn_cache.upstream[18]	PVBusSlave
CCN502.ccn_cache.upstream[19]	PVBusSlave
CCN502.ccn_cache.upstream[1]	PVBusSlave
CCN502.ccn_cache.upstream[20]	PVBusSlave
CCN502.ccn_cache.upstream[21]	PVBusSlave
CCN502.ccn_cache.upstream[22]	PVBusSlave
CCN502.ccn_cache.upstream[23]	PVBusSlave
CCN502.ccn_cache.upstream[24]	PVBusSlave
CCN502.ccn_cache.upstream[25]	PVBusSlave
CCN502.ccn_cache.upstream[26]	PVBusSlave
CCN502.ccn_cache.upstream[27]	PVBusSlave
CCN502.ccn_cache.upstream[28]	PVBusSlave
CCN502.ccn_cache.upstream[29]	PVBusSlave
CCN502.ccn_cache.upstream[2]	PVBusSlave

InstanceName	ComponentName
CCN502.ccn_cache.upstream[30]	PVBusSlave
CCN502.ccn_cache.upstream[31]	PVBusSlave
CCN502.ccn_cache.upstream[32]	PVBusSlave
CCN502.ccn_cache.upstream[33]	PVBusSlave
CCN502.ccn_cache.upstream[34]	PVBusSlave
CCN502.ccn_cache.upstream[35]	PVBusSlave
CCN502.ccn_cache.upstream[36]	PVBusSlave
CCN502.ccn_cache.upstream[37]	PVBusSlave
CCN502.ccn_cache.upstream[38]	PVBusSlave
CCN502.ccn_cache.upstream[39]	PVBusSlave
CCN502.ccn_cache.upstream[3]	PVBusSlave
CCN502.ccn_cache.upstream[40]	PVBusSlave
CCN502.ccn_cache.upstream[41]	PVBusSlave
CCN502.ccn_cache.upstream[42]	PVBusSlave
CCN502.ccn_cache.upstream[43]	PVBusSlave
CCN502.ccn_cache.upstream[44]	PVBusSlave
CCN502.ccn_cache.upstream[45]	PVBusSlave
CCN502.ccn_cache.upstream[46]	PVBusSlave
CCN502.ccn_cache.upstream[47]	PVBusSlave
CCN502.ccn_cache.upstream[4]	PVBusSlave
CCN502.ccn_cache.upstream[5]	PVBusSlave
CCN502.ccn_cache.upstream[6]	PVBusSlave
CCN502.ccn_cache.upstream[7]	PVBusSlave
CCN502.ccn_cache.upstream[8]	PVBusSlave
CCN502.ccn_cache.upstream[9]	PVBusSlave
CCN502.ccn_registers	CCNRegisterSet
CCN502.ccn_registers.bus_slave	PVBusSlave
CCN502.ccn_router	PVBusMapper

This model has the following MTI trace components:

**Table 3-872: CCN502 MTI instances**

InstanceName	ComponentName
CCN502.bus_slave_ocm	PVBusSlave
CCN502.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN502.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN502.ccn_cache	CCNCache
CCN502.ccn_cache.upstream[0]	PVBusSlave
CCN502.ccn_cache.upstream[10]	PVBusSlave
CCN502.ccn_cache.upstream[11]	PVBusSlave

InstanceName	ComponentName
CCN502.ccn_cache.upstream[12]	PVBusSlave
CCN502.ccn_cache.upstream[13]	PVBusSlave
CCN502.ccn_cache.upstream[14]	PVBusSlave
CCN502.ccn_cache.upstream[15]	PVBusSlave
CCN502.ccn_cache.upstream[16]	PVBusSlave
CCN502.ccn_cache.upstream[17]	PVBusSlave
CCN502.ccn_cache.upstream[18]	PVBusSlave
CCN502.ccn_cache.upstream[19]	PVBusSlave
CCN502.ccn_cache.upstream[1]	PVBusSlave
CCN502.ccn_cache.upstream[20]	PVBusSlave
CCN502.ccn_cache.upstream[21]	PVBusSlave
CCN502.ccn_cache.upstream[22]	PVBusSlave
CCN502.ccn_cache.upstream[23]	PVBusSlave
CCN502.ccn_cache.upstream[24]	PVBusSlave
CCN502.ccn_cache.upstream[25]	PVBusSlave
CCN502.ccn_cache.upstream[26]	PVBusSlave
CCN502.ccn_cache.upstream[27]	PVBusSlave
CCN502.ccn_cache.upstream[28]	PVBusSlave
CCN502.ccn_cache.upstream[29]	PVBusSlave
CCN502.ccn_cache.upstream[2]	PVBusSlave
CCN502.ccn_cache.upstream[30]	PVBusSlave
CCN502.ccn_cache.upstream[31]	PVBusSlave
CCN502.ccn_cache.upstream[32]	PVBusSlave
CCN502.ccn_cache.upstream[33]	PVBusSlave
CCN502.ccn_cache.upstream[34]	PVBusSlave
CCN502.ccn_cache.upstream[35]	PVBusSlave
CCN502.ccn_cache.upstream[36]	PVBusSlave
CCN502.ccn_cache.upstream[37]	PVBusSlave
CCN502.ccn_cache.upstream[38]	PVBusSlave
CCN502.ccn_cache.upstream[39]	PVBusSlave
CCN502.ccn_cache.upstream[3]	PVBusSlave
CCN502.ccn_cache.upstream[40]	PVBusSlave
CCN502.ccn_cache.upstream[41]	PVBusSlave
CCN502.ccn_cache.upstream[42]	PVBusSlave
CCN502.ccn_cache.upstream[43]	PVBusSlave
CCN502.ccn_cache.upstream[44]	PVBusSlave
CCN502.ccn_cache.upstream[45]	PVBusSlave
CCN502.ccn_cache.upstream[46]	PVBusSlave
CCN502.ccn_cache.upstream[47]	PVBusSlave

InstanceName	ComponentName
CCN502.ccn_cache.upstream[4]	PVBusSlave
CCN502.ccn_cache.upstream[5]	PVBusSlave
CCN502.ccn_cache.upstream[6]	PVBusSlave
CCN502.ccn_cache.upstream[7]	PVBusSlave
CCN502.ccn_cache.upstream[8]	PVBusSlave
CCN502.ccn_cache.upstream[9]	PVBusSlave
CCN502.ccn_registers	CCNRegisterSet
CCN502.ccn_registers.bus_slave	PVBusSlave
CCN502.ccn_router	PVBusMapper

## Ports for CCN502

**Table 3-873: Ports**

Name	Protocol	Type	Description
pvbus_m_hni[1]	PVBus	Master	HNI downstream port.
pvbus_m_snf[4]	PVBus	Master	SNF downstream ports.
pvbus_s_rnf[4]	PVBus	Slave	RNF upstream ports.
pvbus_s_rni[9]	PVBus	Slave	RNI upstream ports
reset_in	Signal	Slave	Reset signal.

## Parameters for CCN502

### **acchannelen\_rnf**

#### Type

int

#### Default value

0xf

#### Description

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

### **acchannelen\_rni**

#### Type

int

#### Default value

0x1ff

#### Description

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**cache\_size\_in\_kbytes****Type**

int

**Default value**

0x1000

**Description**

Number of kilo bytes in cache.

**cache\_state\_modelled****Type**

bool

**Default value**

0x1

**Description**

Model the cache state.

**ccn\_cache.acchannelen\_rnf****Type**

int

**Default value**

0xf

**Description**

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

**ccn\_cache.acchannelen\_rni****Type**

int

**Default value**

0xffff

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**ccn\_cache.cache\_size\_kb****Type**

int

**Default value**

0x2000

**Description**

Number of kilo bytes in cache.

**ccn\_cache.cache\_state\_modelled****Type**

bool

**Default value**

0x1

**Description**

Model the cache state.

**ccn\_cache.force\_on\_from\_start****Type**

bool

**Default value**

0x0

**Description**

Enables snooping on upstream ports from the start of simulation. The CCI will normally start up with snooping disabled, however, using this parameter we allow the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset. If the upstreams can ever be held in reset then you *\*must\** connect the `reset_state_of_upstream_port_3` and `reset_state_of_upstream_port_4` so that it knows when to disable snoops to the upstream systems. Otherwise, the upstream system will receive snoop messages whilst in reset and it will complain that it 'received a snoop request whilst it was in reset'.

**ccn\_cache.number\_of\_phantom\_entries****Type**

int

**Default value**

0x20

**Description**

Number of phantom entries in the cache. Phantom entries are used by certain cache operations to hold temporary data. Usually this should be left at the default value which is safe for all systems containing up to 32 masters.

**ccn\_cache.periphbase****Type**

int

**Default value**

0x2c000000

**Description**

Value for PERIPHBASE. Only bits [43:24] are used.

**ccn\_cache.reentrancy\_support****Type**

string

**Default value**

env

**Description**

Must be one of: on/off/cacheglobal/env: 'on': hazard checking per cache line (normal mode), 'off': no hazard checking (use only for single master systems), 'cacheglobal': hazard checking globally for cache (not per cache line, testing feature, provokes more hazards than necessary), 'env' (or empty string): take value from FM\_REENTRANCY\_SUPPORT env var, if this is not set use 'on', default is 'env'.

**ccn\_cache.sbas\_bridge\_present****Type**

bool

**Default value**

0x1

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**ccn\_cache.sbsx\_bridge\_present****Type**

bool

**Default value**

0x1

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enable PVBUSLogger for downstream ports.

**force\_on\_from\_start****Type**

bool

**Default value**

false

Enables snooping on upstream ports from the start of simulation.

The CCN502 will normally start up with snooping disabled. However, using this parameter allows the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset

**number\_of\_snf****Type**

int

**Default value**

0x2

**Description**

Number of SNF nodes present.

**periphbase****Type**

int

**Default value**

0x2c000000

**Description**

Value for PERIPHBASE. Only bits [43:24] are used.

**sbsx\_bridge\_present****Type**

bool

**Default value**

0x1

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**systemaddrmap****Type**

int

**Default value**

0x0

**Description**

Bitmap for 20 regions in CCN Interconnect. Every two bits describes the region type for corresponding region.



**variant\_name**

**Type**

string

**Default value**

"CCN502\_6XP"

**Description**

Can be either CCN502\_6XP or CCN502\_8XP.

3.10.9 CCN504

CCN504 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-874: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About CCN504

CCN504 has three downstream ports:

- Two SNF ports for the memory controller
- One Acelite port (HNI)



The parameters for the ccncache subcomponent are not accessible in System Canvas.

Iris and MTI instances for CCN504

This model has the following Iris instances:

Table 3-875: CCN504 Iris instances

InstanceName	ComponentName
CCN504	CCN5XX
CCN504.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN504.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN504.ccn_cache	CCNCache
CCN504.ccn_cache.upstream[0]	PVBusSlave
CCN504.ccn_cache.upstream[10]	PVBusSlave

InstanceName	ComponentName
CCN504.ccn_cache.upstream[11]	PVBusSlave
CCN504.ccn_cache.upstream[12]	PVBusSlave
CCN504.ccn_cache.upstream[13]	PVBusSlave
CCN504.ccn_cache.upstream[14]	PVBusSlave
CCN504.ccn_cache.upstream[15]	PVBusSlave
CCN504.ccn_cache.upstream[16]	PVBusSlave
CCN504.ccn_cache.upstream[17]	PVBusSlave
CCN504.ccn_cache.upstream[18]	PVBusSlave
CCN504.ccn_cache.upstream[19]	PVBusSlave
CCN504.ccn_cache.upstream[1]	PVBusSlave
CCN504.ccn_cache.upstream[20]	PVBusSlave
CCN504.ccn_cache.upstream[21]	PVBusSlave
CCN504.ccn_cache.upstream[22]	PVBusSlave
CCN504.ccn_cache.upstream[23]	PVBusSlave
CCN504.ccn_cache.upstream[24]	PVBusSlave
CCN504.ccn_cache.upstream[25]	PVBusSlave
CCN504.ccn_cache.upstream[26]	PVBusSlave
CCN504.ccn_cache.upstream[27]	PVBusSlave
CCN504.ccn_cache.upstream[28]	PVBusSlave
CCN504.ccn_cache.upstream[29]	PVBusSlave
CCN504.ccn_cache.upstream[2]	PVBusSlave
CCN504.ccn_cache.upstream[30]	PVBusSlave
CCN504.ccn_cache.upstream[31]	PVBusSlave
CCN504.ccn_cache.upstream[32]	PVBusSlave
CCN504.ccn_cache.upstream[33]	PVBusSlave
CCN504.ccn_cache.upstream[34]	PVBusSlave
CCN504.ccn_cache.upstream[35]	PVBusSlave
CCN504.ccn_cache.upstream[36]	PVBusSlave
CCN504.ccn_cache.upstream[37]	PVBusSlave
CCN504.ccn_cache.upstream[38]	PVBusSlave
CCN504.ccn_cache.upstream[39]	PVBusSlave
CCN504.ccn_cache.upstream[3]	PVBusSlave
CCN504.ccn_cache.upstream[40]	PVBusSlave
CCN504.ccn_cache.upstream[41]	PVBusSlave
CCN504.ccn_cache.upstream[42]	PVBusSlave
CCN504.ccn_cache.upstream[43]	PVBusSlave
CCN504.ccn_cache.upstream[44]	PVBusSlave
CCN504.ccn_cache.upstream[45]	PVBusSlave
CCN504.ccn_cache.upstream[46]	PVBusSlave

InstanceName	ComponentName
CCN504.ccn_cache.upstream[47]	PVBusSlave
CCN504.ccn_cache.upstream[4]	PVBusSlave
CCN504.ccn_cache.upstream[5]	PVBusSlave
CCN504.ccn_cache.upstream[6]	PVBusSlave
CCN504.ccn_cache.upstream[7]	PVBusSlave
CCN504.ccn_cache.upstream[8]	PVBusSlave
CCN504.ccn_cache.upstream[9]	PVBusSlave
CCN504.ccn_registers	CCNRegisterSet
CCN504.ccn_registers.bus_slave	PVBusSlave
CCN504.ccn_router	PVBusMapper

This model has the following MTI trace components:

**Table 3-876: CCN504 MTI instances**

InstanceName	ComponentName
CCN504.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN504.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN504.ccn_cache	CCNCache
CCN504.ccn_cache.upstream[0]	PVBusSlave
CCN504.ccn_cache.upstream[10]	PVBusSlave
CCN504.ccn_cache.upstream[11]	PVBusSlave
CCN504.ccn_cache.upstream[12]	PVBusSlave
CCN504.ccn_cache.upstream[13]	PVBusSlave
CCN504.ccn_cache.upstream[14]	PVBusSlave
CCN504.ccn_cache.upstream[15]	PVBusSlave
CCN504.ccn_cache.upstream[16]	PVBusSlave
CCN504.ccn_cache.upstream[17]	PVBusSlave
CCN504.ccn_cache.upstream[18]	PVBusSlave
CCN504.ccn_cache.upstream[19]	PVBusSlave
CCN504.ccn_cache.upstream[1]	PVBusSlave
CCN504.ccn_cache.upstream[20]	PVBusSlave
CCN504.ccn_cache.upstream[21]	PVBusSlave
CCN504.ccn_cache.upstream[22]	PVBusSlave
CCN504.ccn_cache.upstream[23]	PVBusSlave
CCN504.ccn_cache.upstream[24]	PVBusSlave
CCN504.ccn_cache.upstream[25]	PVBusSlave
CCN504.ccn_cache.upstream[26]	PVBusSlave
CCN504.ccn_cache.upstream[27]	PVBusSlave
CCN504.ccn_cache.upstream[28]	PVBusSlave
CCN504.ccn_cache.upstream[29]	PVBusSlave

InstanceName	ComponentName
CCN504.ccn_cache.upstream[2]	PVBusSlave
CCN504.ccn_cache.upstream[30]	PVBusSlave
CCN504.ccn_cache.upstream[31]	PVBusSlave
CCN504.ccn_cache.upstream[32]	PVBusSlave
CCN504.ccn_cache.upstream[33]	PVBusSlave
CCN504.ccn_cache.upstream[34]	PVBusSlave
CCN504.ccn_cache.upstream[35]	PVBusSlave
CCN504.ccn_cache.upstream[36]	PVBusSlave
CCN504.ccn_cache.upstream[37]	PVBusSlave
CCN504.ccn_cache.upstream[38]	PVBusSlave
CCN504.ccn_cache.upstream[39]	PVBusSlave
CCN504.ccn_cache.upstream[3]	PVBusSlave
CCN504.ccn_cache.upstream[40]	PVBusSlave
CCN504.ccn_cache.upstream[41]	PVBusSlave
CCN504.ccn_cache.upstream[42]	PVBusSlave
CCN504.ccn_cache.upstream[43]	PVBusSlave
CCN504.ccn_cache.upstream[44]	PVBusSlave
CCN504.ccn_cache.upstream[45]	PVBusSlave
CCN504.ccn_cache.upstream[46]	PVBusSlave
CCN504.ccn_cache.upstream[47]	PVBusSlave
CCN504.ccn_cache.upstream[4]	PVBusSlave
CCN504.ccn_cache.upstream[5]	PVBusSlave
CCN504.ccn_cache.upstream[6]	PVBusSlave
CCN504.ccn_cache.upstream[7]	PVBusSlave
CCN504.ccn_cache.upstream[8]	PVBusSlave
CCN504.ccn_cache.upstream[9]	PVBusSlave
CCN504.ccn_registers	CCNRegisterSet
CCN504.ccn_registers.bus_slave	PVBusSlave
CCN504.ccn_router	PVBusMapper

## Ports for CCN504

**Table 3-877: Ports**

Name	Protocol	Type	Description
pvbus_m_hni[1]	PVBus	Master	HNI downstream port.
pvbus_m_snf[2]	PVBus	Master	SNF downstream ports.
pvbus_s_rnf[4]	PVBus	Slave	RNF upstream ports.
pvbus_s_rni[18]	PVBus	Slave	RNI upstream ports.
reset_in	Signal	Slave	Reset signal.

## Parameters for CCN504

### **acchannelen\_rnf**

**Type**

int

**Default value**

0xf

**Description**

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

### **acchannelen\_rni**

**Type**

int

**Default value**

0x3ffff

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

### **cache\_size\_in\_mbytes**

**Type**

int

**Default value**

0x8

**Description**

Number of mega bytes in cache.

### **cache\_state\_modelled**

**Type**

bool

**Default value**

0x1

**Description**

Model the cache state.

### **ccn\_cache.acchannelen\_rnf**

**Type**

int

**Default value**

0xf

**Description**

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

**ccn\_cache.acchannelen\_rni**

**Type**

int

**Default value**

0xffff

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**ccn\_cache.cache\_size\_kb**

**Type**

int

**Default value**

0x2000

**Description**

Number of kilo bytes in cache.

**ccn\_cache.cache\_state\_modelled**

**Type**

bool

**Default value**

0x1

**Description**

Model the cache state.

**ccn\_cache.force\_on\_from\_start**

**Type**

bool

**Default value**

0x0

**Description**

Enables snooping on upstream ports from the start of simulation. The CCI will normally start up with snooping disabled, however, using this parameter we allow the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset. If the upstreams can ever be held in reset then you *\*must\** connect the `reset_state_of_upstream_port_3` and `reset_state_of_upstream_port_4` so that it knows when to disable snoops to the upstream systems. Otherwise, the upstream system will receive snoop messages whilst in reset and it will complain that it 'received a snoop request whilst it was in reset'.

**ccn\_cache.number\_of\_phantom\_entries****Type**

int

**Default value**

0x20

**Description**

Number of phantom entries in the cache. Phantom entries are used by certain cache operations to hold temporary data. Usually this should be left at the default value which is safe for all systems containing up to 32 masters.

**ccn\_cache.periphbase****Type**

int

**Default value**

0x2c000000

**Description**

Value for PERIPHBASE. Only bits [43:24] are used.

**ccn\_cache.reentrancy\_support****Type**

string

**Default value**

env

**Description**

Must be one of: on/off/cacheglobal/env: 'on': hazard checking per cache line (normal mode), 'off': no hazard checking (use only for single master systems), 'cacheglobal': hazard checking globally for cache (not per cache line, testing feature, provokes more hazards than necessary), 'env' (or empty string): take value from FM\_REENTRANCY\_SUPPORT env var, if this is not set use 'on', default is 'env'.

**ccn\_cache.sbas\_bridge\_present****Type**

bool

**Default value**

0x1

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**ccn\_cache.sbsx\_bridge\_present****Type**

bool

**Default value**

0x1

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**disable\_hni\_cacheable\_error****Type**

bool

**Default value**

0x0

**Description**

Disable sending error response when HNI receives cacheable access.

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enable PVBUSLoggers for the downstream ports in the CCN model.

**force\_on\_from\_start****Type**

bool

**Default value**

false

Enables snooping on upstream ports from the start of simulation.

The CCN504 will normally start up with snooping disabled. However, using this parameter allows the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset.

**number\_of\_snf****Type**

int

**Default value**

0x2



**Description**

Number of SNF nodes present.

**periphbase****Type**

int

**Default value**

0x2c000000

**Description**

Value for PERIPHBASE. Only bits [43:24] are used.

**sbas\_bridge\_present****Type**

bool

**Default value**

0x1

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**sbsx\_bridge\_present****Type**

bool

**Default value**

0x1

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**systemaddrmap****Type**

int

**Default value**

0x0

**Description**

Bitmap for 20 regions in CCN Interconnect. Every two bits describes the region type for corresponding region.

### 3.10.10 CCN508

CCN508 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-878: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About CCN508

CCN508 has six downstream ports:

- Four SNF ports for the memory controller.
- Two Acelite ports (HNI).



Note

The parameters for the CCNCache subcomponent are not accessible in System Canvas.

#### Iris and MTI instances for CCN508

This model has the following Iris instances:

**Table 3-879: CCN508 Iris instances**

InstanceName	ComponentName
CCN508	CCN5XX
CCN508.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN508.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN508.ccn502_hni_exclusive_monitor_1	PVBusExclusiveMonitor
CCN508.ccn502_hni_exclusive_monitor_1.bus_mapper	PVBusMapper
CCN508.ccn_cache	CCNCache
CCN508.ccn_cache.upstream[0]	PVBusSlave
CCN508.ccn_cache.upstream[10]	PVBusSlave
CCN508.ccn_cache.upstream[11]	PVBusSlave
CCN508.ccn_cache.upstream[12]	PVBusSlave
CCN508.ccn_cache.upstream[13]	PVBusSlave
CCN508.ccn_cache.upstream[14]	PVBusSlave
CCN508.ccn_cache.upstream[15]	PVBusSlave
CCN508.ccn_cache.upstream[16]	PVBusSlave
CCN508.ccn_cache.upstream[17]	PVBusSlave

InstanceName	ComponentName
CCN508.ccn_cache.upstream[18]	PVBusSlave
CCN508.ccn_cache.upstream[19]	PVBusSlave
CCN508.ccn_cache.upstream[1]	PVBusSlave
CCN508.ccn_cache.upstream[20]	PVBusSlave
CCN508.ccn_cache.upstream[21]	PVBusSlave
CCN508.ccn_cache.upstream[22]	PVBusSlave
CCN508.ccn_cache.upstream[23]	PVBusSlave
CCN508.ccn_cache.upstream[24]	PVBusSlave
CCN508.ccn_cache.upstream[25]	PVBusSlave
CCN508.ccn_cache.upstream[26]	PVBusSlave
CCN508.ccn_cache.upstream[27]	PVBusSlave
CCN508.ccn_cache.upstream[28]	PVBusSlave
CCN508.ccn_cache.upstream[29]	PVBusSlave
CCN508.ccn_cache.upstream[2]	PVBusSlave
CCN508.ccn_cache.upstream[30]	PVBusSlave
CCN508.ccn_cache.upstream[31]	PVBusSlave
CCN508.ccn_cache.upstream[32]	PVBusSlave
CCN508.ccn_cache.upstream[33]	PVBusSlave
CCN508.ccn_cache.upstream[34]	PVBusSlave
CCN508.ccn_cache.upstream[35]	PVBusSlave
CCN508.ccn_cache.upstream[36]	PVBusSlave
CCN508.ccn_cache.upstream[37]	PVBusSlave
CCN508.ccn_cache.upstream[38]	PVBusSlave
CCN508.ccn_cache.upstream[39]	PVBusSlave
CCN508.ccn_cache.upstream[3]	PVBusSlave
CCN508.ccn_cache.upstream[40]	PVBusSlave
CCN508.ccn_cache.upstream[41]	PVBusSlave
CCN508.ccn_cache.upstream[42]	PVBusSlave
CCN508.ccn_cache.upstream[43]	PVBusSlave
CCN508.ccn_cache.upstream[44]	PVBusSlave
CCN508.ccn_cache.upstream[45]	PVBusSlave
CCN508.ccn_cache.upstream[46]	PVBusSlave
CCN508.ccn_cache.upstream[47]	PVBusSlave
CCN508.ccn_cache.upstream[4]	PVBusSlave
CCN508.ccn_cache.upstream[5]	PVBusSlave
CCN508.ccn_cache.upstream[6]	PVBusSlave
CCN508.ccn_cache.upstream[7]	PVBusSlave
CCN508.ccn_cache.upstream[8]	PVBusSlave
CCN508.ccn_cache.upstream[9]	PVBusSlave

InstanceName	ComponentName
CCN508.ccn_registers	CCNRegisterSet
CCN508.ccn_registers.bus_slave	PVBusSlave
CCN508.ccn_router	PVBusMapper

This model has the following MTI trace components:

**Table 3-880: CCN508 MTI instances**

InstanceName	ComponentName
CCN508.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN508.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN508.ccn502_hni_exclusive_monitor_1	PVBusExclusiveMonitor
CCN508.ccn502_hni_exclusive_monitor_1.bus_mapper	PVBusMapper
CCN508.ccn_cache	CCNCache
CCN508.ccn_cache.upstream[0]	PVBusSlave
CCN508.ccn_cache.upstream[10]	PVBusSlave
CCN508.ccn_cache.upstream[11]	PVBusSlave
CCN508.ccn_cache.upstream[12]	PVBusSlave
CCN508.ccn_cache.upstream[13]	PVBusSlave
CCN508.ccn_cache.upstream[14]	PVBusSlave
CCN508.ccn_cache.upstream[15]	PVBusSlave
CCN508.ccn_cache.upstream[16]	PVBusSlave
CCN508.ccn_cache.upstream[17]	PVBusSlave
CCN508.ccn_cache.upstream[18]	PVBusSlave
CCN508.ccn_cache.upstream[19]	PVBusSlave
CCN508.ccn_cache.upstream[1]	PVBusSlave
CCN508.ccn_cache.upstream[20]	PVBusSlave
CCN508.ccn_cache.upstream[21]	PVBusSlave
CCN508.ccn_cache.upstream[22]	PVBusSlave
CCN508.ccn_cache.upstream[23]	PVBusSlave
CCN508.ccn_cache.upstream[24]	PVBusSlave
CCN508.ccn_cache.upstream[25]	PVBusSlave
CCN508.ccn_cache.upstream[26]	PVBusSlave
CCN508.ccn_cache.upstream[27]	PVBusSlave
CCN508.ccn_cache.upstream[28]	PVBusSlave
CCN508.ccn_cache.upstream[29]	PVBusSlave
CCN508.ccn_cache.upstream[2]	PVBusSlave
CCN508.ccn_cache.upstream[30]	PVBusSlave
CCN508.ccn_cache.upstream[31]	PVBusSlave
CCN508.ccn_cache.upstream[32]	PVBusSlave
CCN508.ccn_cache.upstream[33]	PVBusSlave

InstanceName	ComponentName
CCN508.ccn_cache.upstream[34]	PVBusSlave
CCN508.ccn_cache.upstream[35]	PVBusSlave
CCN508.ccn_cache.upstream[36]	PVBusSlave
CCN508.ccn_cache.upstream[37]	PVBusSlave
CCN508.ccn_cache.upstream[38]	PVBusSlave
CCN508.ccn_cache.upstream[39]	PVBusSlave
CCN508.ccn_cache.upstream[3]	PVBusSlave
CCN508.ccn_cache.upstream[40]	PVBusSlave
CCN508.ccn_cache.upstream[41]	PVBusSlave
CCN508.ccn_cache.upstream[42]	PVBusSlave
CCN508.ccn_cache.upstream[43]	PVBusSlave
CCN508.ccn_cache.upstream[44]	PVBusSlave
CCN508.ccn_cache.upstream[45]	PVBusSlave
CCN508.ccn_cache.upstream[46]	PVBusSlave
CCN508.ccn_cache.upstream[47]	PVBusSlave
CCN508.ccn_cache.upstream[4]	PVBusSlave
CCN508.ccn_cache.upstream[5]	PVBusSlave
CCN508.ccn_cache.upstream[6]	PVBusSlave
CCN508.ccn_cache.upstream[7]	PVBusSlave
CCN508.ccn_cache.upstream[8]	PVBusSlave
CCN508.ccn_cache.upstream[9]	PVBusSlave
CCN508.ccn_registers	CCNRegisterSet
CCN508.ccn_registers.bus_slave	PVBusSlave
CCN508.ccn_router	PVBusMapper

## Ports for CCN508

**Table 3-881: Ports**

Name	Protocol	Type	Description
pdbus_m_hni[2]	PVBus	Master	HNI downstream ports.
pdbus_m_snf[4]	PVBus	Master	SNF downstream ports.
pdbus_s_rnf[8]	PVBus	Slave	RNF upstream ports.
pdbus_s_rni[24]	PVBus	Slave	RNI upstream ports.
reset_in	Signal	Slave	Reset signal.

## Parameters for CCN508

### acchannelen\_rnf

#### Type

int

**Default value**

0xf

**Description**

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

**acchannelen\_rni****Type**

int

**Default value**

0x3ffff

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**cache\_size\_in\_mbytes****Type**

int

**Default value**

0x8

**Description**

Number of mega bytes in cache.

**cache\_state\_modelled****Type**

bool

**Default value**

0x1

**Description**

Model the cache state.

**ccn\_cache.acchannelen\_rnf****Type**

int

**Default value**

0xf

**Description**

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

**ccn\_cache.acchannelen\_rni****Type**

int

**Default value**

0xffff

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**ccn\_cache.cache\_size\_kb****Type**

int

**Default value**

0x2000

**Description**

Number of kilo bytes in cache.

**ccn\_cache.cache\_state\_modelled****Type**

bool

**Default value**

0x1

**Description**

Model the cache state.

**ccn\_cache.force\_on\_from\_start****Type**

bool

**Default value**

0x0

**Description**

Enables snooping on upstream ports from the start of simulation. The CCI will normally start up with snooping disabled, however, using this parameter we allow the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset. If the upstreams can ever be held in reset then you *\*must\** connect the reset\_state\_of\_upstream\_port\_3 and reset\_state\_of\_upstream\_port\_4 so that it knows when to disable snoops to the upstream systems. Otherwise, the upstream system will receive snoop messages whilst in reset and it will complain that it 'received a snoop request whilst it was in reset'.

**ccn\_cache.number\_of\_phantom\_entries****Type**

int

**Default value**

0x20

**Description**

Number of phantom entries in the cache. Phantom entries are used by certain cache operations to hold temporary data. Usually this should be left at the default value which is safe for all systems containing up to 32 masters.

**ccn\_cache.periphbase****Type**

int

**Default value**

0x2c000000

**Description**

Value for PERIPHBASE. Only bits [43:24] are used.

**ccn\_cache.reentrancy\_support****Type**

string

**Default value**

env

**Description**

Must be one of: on/off/cacheglobal/env: 'on': hazard checking per cache line (normal mode), 'off': no hazard checking (use only for single master systems), 'cacheglobal': hazard checking globally for cache (not per cache line, testing feature, provokes more hazards than necessary), 'env' (or empty string): take value from FM\_REENTRANCY\_SUPPORT env var, if this is not set use 'on', default is 'env'.

**ccn\_cache.sbas\_bridge\_present****Type**

bool

**Default value**

0x1

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**ccn\_cache.sbsx\_bridge\_present****Type**

bool

**Default value**

0x1

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.



**disable\_hni\_cacheable\_error****Type**

bool

**Default value**

0x0

**Description**

Disable sending error response when HNI receives cacheable access.

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enable PVBUSLoggers for the downstream ports in the CCN model.

**force\_on\_from\_start****Type**

bool

**Default value**

false

Enables snooping on upstream ports from the start of simulation.

The CCN508 will normally start up with snooping disabled. However, using this parameter allows the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset.

**number\_of\_snf****Type**

int

**Default value**

0x2

**Description**

Number of SNF nodes present.

**periphbase****Type**

int

**Default value**

0x2c000000

**Description**  
Value for PERIPHBASE. Only bits [43:24] are used.

**sbas\_bridge\_present**

**Type**  
bool

**Default value**  
0x1

**Description**  
For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**sbsx\_bridge\_present**

**Type**  
bool

**Default value**  
0x1

**Description**  
For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**systemaddrmap**

**Type**  
int

**Default value**  
0x0

**Description**  
Bitmap for 20 regions in CCN Interconnect. Every two bits describes the region type for corresponding region.

3.10.11 CCN512

CCN512 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-882: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About CCN512

CCN512 has an L3 cache that can provide coherency between up to 12 fully-coherent ACE clusters and 24 I/O coherent masters. It can connect up to four memory elements to drive transaction requests.

CCN512 has six downstream ports:

- Four SNF ports for the memory controller
- Two Acelite ports (HNI)

Some useful parameters are:

- `cache_state_modelled`
- `cache_size_in_mbytes`
- `systemaddrmap`

## Limitations

The model has the following limitations:

- No support for 3 SN striping.
- If there are multiple SN-Fs, the distribution of addresses to each SN-F is not guaranteed to match the hardware.
- The parameters for the CCNCache subcomponent are not accessible in System Canvas.

## Iris and MTI instances for CCN512

This model has the following Iris instances:

**Table 3-883: CCN512 Iris instances**

InstanceName	ComponentName
CCN512	CCN5XX
CCN512.bus_slave_ocm	PVBusSlave
CCN512.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN512.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN512.ccn502_hni_exclusive_monitor_1	PVBusExclusiveMonitor
CCN512.ccn502_hni_exclusive_monitor_1.bus_mapper	PVBusMapper
CCN512.ccn_cache	CCNCache
CCN512.ccn_cache.upstream[0]	PVBusSlave
CCN512.ccn_cache.upstream[10]	PVBusSlave
CCN512.ccn_cache.upstream[11]	PVBusSlave
CCN512.ccn_cache.upstream[12]	PVBusSlave
CCN512.ccn_cache.upstream[13]	PVBusSlave
CCN512.ccn_cache.upstream[14]	PVBusSlave
CCN512.ccn_cache.upstream[15]	PVBusSlave
CCN512.ccn_cache.upstream[16]	PVBusSlave

InstanceName	ComponentName
CCN512.ccn_cache.upstream[17]	PVBusSlave
CCN512.ccn_cache.upstream[18]	PVBusSlave
CCN512.ccn_cache.upstream[19]	PVBusSlave
CCN512.ccn_cache.upstream[1]	PVBusSlave
CCN512.ccn_cache.upstream[20]	PVBusSlave
CCN512.ccn_cache.upstream[21]	PVBusSlave
CCN512.ccn_cache.upstream[22]	PVBusSlave
CCN512.ccn_cache.upstream[23]	PVBusSlave
CCN512.ccn_cache.upstream[24]	PVBusSlave
CCN512.ccn_cache.upstream[25]	PVBusSlave
CCN512.ccn_cache.upstream[26]	PVBusSlave
CCN512.ccn_cache.upstream[27]	PVBusSlave
CCN512.ccn_cache.upstream[28]	PVBusSlave
CCN512.ccn_cache.upstream[29]	PVBusSlave
CCN512.ccn_cache.upstream[2]	PVBusSlave
CCN512.ccn_cache.upstream[30]	PVBusSlave
CCN512.ccn_cache.upstream[31]	PVBusSlave
CCN512.ccn_cache.upstream[32]	PVBusSlave
CCN512.ccn_cache.upstream[33]	PVBusSlave
CCN512.ccn_cache.upstream[34]	PVBusSlave
CCN512.ccn_cache.upstream[35]	PVBusSlave
CCN512.ccn_cache.upstream[36]	PVBusSlave
CCN512.ccn_cache.upstream[37]	PVBusSlave
CCN512.ccn_cache.upstream[38]	PVBusSlave
CCN512.ccn_cache.upstream[39]	PVBusSlave
CCN512.ccn_cache.upstream[3]	PVBusSlave
CCN512.ccn_cache.upstream[40]	PVBusSlave
CCN512.ccn_cache.upstream[41]	PVBusSlave
CCN512.ccn_cache.upstream[42]	PVBusSlave
CCN512.ccn_cache.upstream[43]	PVBusSlave
CCN512.ccn_cache.upstream[44]	PVBusSlave
CCN512.ccn_cache.upstream[45]	PVBusSlave
CCN512.ccn_cache.upstream[46]	PVBusSlave
CCN512.ccn_cache.upstream[47]	PVBusSlave
CCN512.ccn_cache.upstream[4]	PVBusSlave
CCN512.ccn_cache.upstream[5]	PVBusSlave
CCN512.ccn_cache.upstream[6]	PVBusSlave
CCN512.ccn_cache.upstream[7]	PVBusSlave
CCN512.ccn_cache.upstream[8]	PVBusSlave

InstanceName	ComponentName
CCN512.ccn_cache.upstream[9]	PVBusSlave
CCN512.ccn_registers	CCNRegisterSet
CCN512.ccn_registers.bus_slave	PVBusSlave
CCN512.ccn_router	PVBusMapper

This model has the following MTI trace components:

**Table 3-884: CCN512 MTI instances**

InstanceName	ComponentName
CCN512.bus_slave_ocm	PVBusSlave
CCN512.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN512.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN512.ccn502_hni_exclusive_monitor_1	PVBusExclusiveMonitor
CCN512.ccn502_hni_exclusive_monitor_1.bus_mapper	PVBusMapper
CCN512.ccn_cache	CCNCache
CCN512.ccn_cache.upstream[0]	PVBusSlave
CCN512.ccn_cache.upstream[10]	PVBusSlave
CCN512.ccn_cache.upstream[11]	PVBusSlave
CCN512.ccn_cache.upstream[12]	PVBusSlave
CCN512.ccn_cache.upstream[13]	PVBusSlave
CCN512.ccn_cache.upstream[14]	PVBusSlave
CCN512.ccn_cache.upstream[15]	PVBusSlave
CCN512.ccn_cache.upstream[16]	PVBusSlave
CCN512.ccn_cache.upstream[17]	PVBusSlave
CCN512.ccn_cache.upstream[18]	PVBusSlave
CCN512.ccn_cache.upstream[19]	PVBusSlave
CCN512.ccn_cache.upstream[1]	PVBusSlave
CCN512.ccn_cache.upstream[20]	PVBusSlave
CCN512.ccn_cache.upstream[21]	PVBusSlave
CCN512.ccn_cache.upstream[22]	PVBusSlave
CCN512.ccn_cache.upstream[23]	PVBusSlave
CCN512.ccn_cache.upstream[24]	PVBusSlave
CCN512.ccn_cache.upstream[25]	PVBusSlave
CCN512.ccn_cache.upstream[26]	PVBusSlave
CCN512.ccn_cache.upstream[27]	PVBusSlave
CCN512.ccn_cache.upstream[28]	PVBusSlave
CCN512.ccn_cache.upstream[29]	PVBusSlave
CCN512.ccn_cache.upstream[2]	PVBusSlave
CCN512.ccn_cache.upstream[30]	PVBusSlave
CCN512.ccn_cache.upstream[31]	PVBusSlave

InstanceName	ComponentName
CCN512.ccn_cache.upstream[32]	PVBusSlave
CCN512.ccn_cache.upstream[33]	PVBusSlave
CCN512.ccn_cache.upstream[34]	PVBusSlave
CCN512.ccn_cache.upstream[35]	PVBusSlave
CCN512.ccn_cache.upstream[36]	PVBusSlave
CCN512.ccn_cache.upstream[37]	PVBusSlave
CCN512.ccn_cache.upstream[38]	PVBusSlave
CCN512.ccn_cache.upstream[39]	PVBusSlave
CCN512.ccn_cache.upstream[3]	PVBusSlave
CCN512.ccn_cache.upstream[40]	PVBusSlave
CCN512.ccn_cache.upstream[41]	PVBusSlave
CCN512.ccn_cache.upstream[42]	PVBusSlave
CCN512.ccn_cache.upstream[43]	PVBusSlave
CCN512.ccn_cache.upstream[44]	PVBusSlave
CCN512.ccn_cache.upstream[45]	PVBusSlave
CCN512.ccn_cache.upstream[46]	PVBusSlave
CCN512.ccn_cache.upstream[47]	PVBusSlave
CCN512.ccn_cache.upstream[4]	PVBusSlave
CCN512.ccn_cache.upstream[5]	PVBusSlave
CCN512.ccn_cache.upstream[6]	PVBusSlave
CCN512.ccn_cache.upstream[7]	PVBusSlave
CCN512.ccn_cache.upstream[8]	PVBusSlave
CCN512.ccn_cache.upstream[9]	PVBusSlave
CCN512.ccn_registers	CCNRegisterSet
CCN512.ccn_registers.bus_slave	PVBusSlave
CCN512.ccn_router	PVBusMapper

## Ports for CCN512

**Table 3-885: Ports**

Name	Protocol	Type	Description
pdbus_m_hni[2]	PVBus	Master	HNI downstream ports.
pdbus_m_snf[4]	PVBus	Master	SNF downstream ports.
pdbus_s_rnf[12]	PVBus	Slave	RNF upstream ports.
pdbus_s_rni[24]	PVBus	Slave	RNI upstream ports.
reset_in	Signal	Slave	Reset signal.

## Parameters for CCN512

### **acchannelen\_rnf**

**Type**

int

**Default value**

0xfff

**Description**

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

### **acchannelen\_rni**

**Type**

int

**Default value**

0xffffffff

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

### **cache\_size\_in\_mbytes**

**Type**

int

**Default value**

0x8

**Description**

Number of mega bytes in cache.

### **cache\_state\_modelled**

**Type**

bool

**Default value**

0x1

**Description**

Model the cache state.

### **ccn\_cache.acchannelen\_rnf**

**Type**

int

**Default value**

0xf

**Description**

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

**ccn\_cache.acchannelen\_rni**

**Type**

int

**Default value**

0xffff

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**ccn\_cache.cache\_size\_kb**

**Type**

int

**Default value**

0x2000

**Description**

Number of kilo bytes in cache.

**ccn\_cache.cache\_state\_modelled**

**Type**

bool

**Default value**

0x1

**Description**

Model the cache state.

**ccn\_cache.force\_on\_from\_start**

**Type**

bool

**Default value**

0x0

**Description**

Enables snooping on upstream ports from the start of simulation. The CCI will normally start up with snooping disabled, however, using this parameter we allow the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset. If the upstreams can ever be held in reset then you *\*must\** connect the reset\_state\_of\_upstream\_port\_3 and reset\_state\_of\_upstream\_port\_4 so that it knows when to disable snoops to the upstream systems. Otherwise, the upstream system will receive snoop messages whilst in reset and it will complain that it 'received a snoop request whilst it was in reset'.



**ccn\_cache.number\_of\_phantom\_entries****Type**

int

**Default value**

0x20

**Description**

Number of phantom entries in the cache. Phantom entries are used by certain cache operations to hold temporary data. Usually this should be left at the default value which is safe for all systems containing up to 32 masters.

**ccn\_cache.periphbase****Type**

int

**Default value**

0x2c000000

**Description**

Value for PERIPHBASE. Only bits [43:24] are used.

**ccn\_cache.reentrancy\_support****Type**

string

**Default value**

env

**Description**

Must be one of: on/off/cacheglobal/env: 'on': hazard checking per cache line (normal mode), 'off': no hazard checking (use only for single master systems), 'cacheglobal': hazard checking globally for cache (not per cache line, testing feature, provokes more hazards than necessary), 'env' (or empty string): take value from FM\_REENTRANCY\_SUPPORT env var, if this is not set use 'on', default is 'env'.

**ccn\_cache.sbas\_bridge\_present****Type**

bool

**Default value**

0x1

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**ccn\_cache.sbsx\_bridge\_present****Type**

bool

**Default value**

0x1

**Description**

For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enbale PVBusLogger for downstream ports.

**force\_on\_from\_start****Type**

bool

**Default value**

false

Enables snooping on upstream ports from the start of simulation.

The CCN512 will normally start up with snooping disabled. However, using this parameter allows the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset.

**number\_of\_snf****Type**

int

**Default value**

0x2

**Description**

Number of SNF nodes present.

**periphbase****Type**

int

**Default value**

0x2c000000

**Description**  
Value for PERIPHBASE. Only bits [43:24] are used.

**sbsx\_bridge\_present**

**Type**  
bool

**Default value**  
0x1

**Description**  
For each upstream port, determine if it is enabled or not w.r.t. dvm requests.

**systemaddrmap**

**Type**  
int

**Default value**  
0x0

**Description**  
Bitmap for 20 regions in CCN Interconnect. Every two bits describes the region type for corresponding region.

3.10.12 CHBCR

CXL Host Bridge Component Registers. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-886: IP revisions support

Revision	Quality level
0.0	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Iris and MTI instances for CHBCR

This model has the following Iris instances:

Table 3-887: CHBCR Iris instances

InstanceName	ComponentName
CHBCR	CHBCR

## Ports for CHBCR

**Table 3-888: Ports**

Name	Protocol	Type	Description
apb_bus_s	PVBus	Slave	-
chbcr_map_interrupt_out	Signal	Master	-

## Parameters for CHBCR

### **diagnostics**

#### **Type**

int

#### **Default value**

0x2

#### **Description**

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

### **num\_decoders**

#### **Type**

int

#### **Default value**

0x10

#### **Description**

Number of Decoders.

### **num\_ports**

#### **Type**

int

#### **Default value**

0x4

#### **Description**

Number of target ports.

## 3.10.13 CI700

CI700 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-889: IP revisions support**

Revision	Quality level
r2p0	Full support
r1p0	Full support
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About CI700

- Major IP revisions (rX) are modeled and are controlled by the `version` parameter in the topology file. If topology doesn't contain this parameter, then model param 'revision' is used.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of the CHI nodes. Set it to the name of the yaml configuration file emitted by Socrates. You must use version r1p6-00rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.

The model supports the following features:

- `rnf`, `rni/rnd`, `hni`, and `snf/sbsx` interface ports. The mapping between the port number and `NodeId` is based on the `NodeId` index. For example, `RNF2` controls `pvbuss_s_rnf[2]`. Its index is specified in the `node_info` register as `logical_id`. Similar behavior can be expected for `HN-I`.

If both `RN-D` and `RN-I` nodes are present, then all starting `rni` ports are mapped to `RN-D` nodes and then the `RN-I` nodes. For example, with two `RN-D` nodes, one `RN-I` node, and given each `RN-I` or `RN-D` node controls three interface ports, `pvbuss_s_rni[0-2]` maps to `RND0`, `pvbuss_s_rni[3-5]` maps to `RND1` and `pvbuss_s_rni[6-8]` maps to `RNI0`.

Similarly, `SN-F` and `SBSX` nodes are mapped to `pvbuss_m_snf[]` ports where starting ports are mapped to `SN-F` and then `SBSX` nodes.

- Mapping `HN*_SLC_SIZE_PARAM` values to cache sizes:

-8

OKB where `HN*_SLC_NUM_WAYS_PARAM=16`



Note

This value is not supported by the model.

-2

128KB where `HN*_SLC_NUM_WAYS_PARAM=16`

**-1**256KB where `HN*_SLC_NUM_WAYS_PARAM=16`**0**512KB where `HN*_SLC_NUM_WAYS_PARAM=16`**1**1MB where `HN*_SLC_NUM_WAYS_PARAM=16`**2**2MB where `HN*_SLC_NUM_WAYS_PARAM=16`**3**3MB where `HN*_SLC_NUM_WAYS_PARAM=12`**3**4MB where `HN*_SLC_NUM_WAYS_PARAM=16`

- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
  - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.
  - When `HNF_ABF_PR.abf_mode` is Reserved and `HNF_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
  - Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
  - The optional interrupt `INTREQPPU` is not supported.

## Model limitations

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
  - Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `por_hnf_ppu_pwpr`. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.

- Non-XY routing behavior is not supported.
- The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.
- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-lite port have not been tested.
- RNSAMs external to the mesh network are not functionally supported.



Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31]) of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address [MAX:12] instead of the actual address [MAX:6], due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address [MAX:6], but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.
- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.
- SLC/LCC partitioning dynamic allocation is not supported.
- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mvp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0
0x00000002000 - 0x00000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- Transactions to addresses unused by Device registers in the Configuration Register Space are always routed to the Configuration Space even when the SAM is configured to use those addresses.
- GIC communication over A4S ports is not supported.

- MPAM features are not supported in the model, and the configuration-dependent register reset values do not match the RTL. Some of the registers (`por_hnf_mpam_idr` to `por_hnf_mpam_mbwumon_idr`) shared between security modes are not present in MPAM\_S.
- The MPAM reset values described in `por_hnf_mpam_ns_por_hnf_mpam_idr[31:24]` default to the not-supported state. The model parameter `hnf_mpam_idr_override` can be used to override this but does not implement any of the functionality.
- MPAM\_S `secure_register_groups_override` is not supported.
- MTU `secure_register_groups_override` is not supported.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.
- For Power State Transition related flush/disable of the SLC due to HNF\_PPU\_PWPR register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.
- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` are used as the default target only when `RNSAM_STATUS.use_default_node=1`. Otherwise, when a txn is sent into the model that does not belong to any of the address regions programmed in the RNSAM, it is routed to the HN-D irrespective of what is programmed in the `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` fields.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- For an SCG, the model does not consider the secondary region if the primary region is not valid.
- HN-D is only permitted on device port P2 in a single-MXP configuration.
- No support for RAS.
- Only the last RN-F in the mesh, which is the one with the highest logical id, can be controlled by `mxp_p[0-5]_syscoreq_ctl` registers. Also, incorrectly, it can be controlled from any XP.
- HN-F SAM limitations:
  - Address masking in default hash regions in HN-F SAM is not supported.
  - Hashing in default hash regions in HN-F SAM is not supported. However, trace `SNF_HASHING_Target_SNF` displays the target SNF as if 3SN/6SN hashing were enabled.



**Note**

The transaction is always routed to the SNO nodeid programmed in the SAM\_CONTROL register.

- Re-programming regions in HN-F SAM is not tested.
- POR\_MTU\_TAG\_ADDR\_CTL.memory\_map\_mode=3'b000 (Pass-through) is the only supported behavior. Other values for this field are not supported.
- For RN-D nodes, when software writes SYSCOREQ, DVM propagation gets enabled but SYSCOACK is not set.

### About the debug\_force\_snoop parameter

The interconnect model normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream `rnf` ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



This parameter is for debug purposes only. Do not use it instead of correctly configuring the model.

## Iris and MTI instances for CI700

This model has the following Iris instances:

**Table 3-890: CI700 Iris instances**

InstanceName	ComponentName
CI700	CI700
CI700.bus_slave_ocm_NS	PVBusSlave
CI700.bus_slave_ocm_S	PVBusSlave
CI700.ci700_tag_cache	CMN_TAG_CACHE
CI700.ci700_tag_cache.metadata_controller0	MetaDataController
CI700.ci700_tag_cache.metadata_controller0.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller1	MetaDataController
CI700.ci700_tag_cache.metadata_controller1.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller10	MetaDataController
CI700.ci700_tag_cache.metadata_controller10.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller100	MetaDataController
CI700.ci700_tag_cache.metadata_controller100.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller101	MetaDataController
CI700.ci700_tag_cache.metadata_controller101.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller102	MetaDataController
CI700.ci700_tag_cache.metadata_controller102.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller103	MetaDataController
CI700.ci700_tag_cache.metadata_controller103.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller104	MetaDataController
CI700.ci700_tag_cache.metadata_controller104.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller105	MetaDataController
CI700.ci700_tag_cache.metadata_controller105.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller106	MetaDataController
CI700.ci700_tag_cache.metadata_controller106.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller107	MetaDataController
CI700.ci700_tag_cache.metadata_controller107.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller108	MetaDataController
CI700.ci700_tag_cache.metadata_controller108.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller109	MetaDataController
CI700.ci700_tag_cache.metadata_controller109.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller11	MetaDataController
CI700.ci700_tag_cache.metadata_controller11.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller110	MetaDataController
CI700.ci700_tag_cache.metadata_controller110.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller111	MetaDataController
CI700.ci700_tag_cache.metadata_controller111.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller112	MetaDataController
CI700.ci700_tag_cache.metadata_controller112.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller113	MetaDataController
CI700.ci700_tag_cache.metadata_controller113.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller114	MetaDataController
CI700.ci700_tag_cache.metadata_controller114.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller115	MetaDataController
CI700.ci700_tag_cache.metadata_controller115.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller116	MetaDataController
CI700.ci700_tag_cache.metadata_controller116.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller117	MetaDataController
CI700.ci700_tag_cache.metadata_controller117.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller118	MetaDataController
CI700.ci700_tag_cache.metadata_controller118.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller119	MetaDataController
CI700.ci700_tag_cache.metadata_controller119.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller12	MetaDataController
CI700.ci700_tag_cache.metadata_controller12.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller120	MetaDataController
CI700.ci700_tag_cache.metadata_controller120.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller121	MetaDataController
CI700.ci700_tag_cache.metadata_controller121.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller122	MetaDataController
CI700.ci700_tag_cache.metadata_controller122.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller123	MetaDataController
CI700.ci700_tag_cache.metadata_controller123.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller124	MetaDataController
CI700.ci700_tag_cache.metadata_controller124.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller125	MetaDataController
CI700.ci700_tag_cache.metadata_controller125.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller126	MetaDataController
CI700.ci700_tag_cache.metadata_controller126.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller127	MetaDataController

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller127.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller13	MetaDataController
CI700.ci700_tag_cache.metadata_controller13.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller14	MetaDataController
CI700.ci700_tag_cache.metadata_controller14.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller15	MetaDataController
CI700.ci700_tag_cache.metadata_controller15.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller16	MetaDataController
CI700.ci700_tag_cache.metadata_controller16.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller17	MetaDataController
CI700.ci700_tag_cache.metadata_controller17.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller18	MetaDataController
CI700.ci700_tag_cache.metadata_controller18.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller19	MetaDataController
CI700.ci700_tag_cache.metadata_controller19.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller2	MetaDataController
CI700.ci700_tag_cache.metadata_controller2.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller20	MetaDataController
CI700.ci700_tag_cache.metadata_controller20.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller21	MetaDataController
CI700.ci700_tag_cache.metadata_controller21.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller22	MetaDataController
CI700.ci700_tag_cache.metadata_controller22.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller23	MetaDataController
CI700.ci700_tag_cache.metadata_controller23.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller24	MetaDataController
CI700.ci700_tag_cache.metadata_controller24.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller25	MetaDataController
CI700.ci700_tag_cache.metadata_controller25.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller26	MetaDataController
CI700.ci700_tag_cache.metadata_controller26.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller27	MetaDataController
CI700.ci700_tag_cache.metadata_controller27.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller28	MetaDataController
CI700.ci700_tag_cache.metadata_controller28.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller29	MetaDataController
CI700.ci700_tag_cache.metadata_controller29.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller3	MetaDataController
CI700.ci700_tag_cache.metadata_controller3.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller30	MetaDataController
CI700.ci700_tag_cache.metadata_controller30.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller31	MetaDataController
CI700.ci700_tag_cache.metadata_controller31.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller32	MetaDataController
CI700.ci700_tag_cache.metadata_controller32.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller33	MetaDataController
CI700.ci700_tag_cache.metadata_controller33.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller34	MetaDataController
CI700.ci700_tag_cache.metadata_controller34.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller35	MetaDataController
CI700.ci700_tag_cache.metadata_controller35.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller36	MetaDataController
CI700.ci700_tag_cache.metadata_controller36.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller37	MetaDataController
CI700.ci700_tag_cache.metadata_controller37.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller38	MetaDataController
CI700.ci700_tag_cache.metadata_controller38.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller39	MetaDataController
CI700.ci700_tag_cache.metadata_controller39.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller4	MetaDataController
CI700.ci700_tag_cache.metadata_controller4.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller40	MetaDataController
CI700.ci700_tag_cache.metadata_controller40.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller41	MetaDataController
CI700.ci700_tag_cache.metadata_controller41.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller42	MetaDataController
CI700.ci700_tag_cache.metadata_controller42.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller43	MetaDataController
CI700.ci700_tag_cache.metadata_controller43.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller44	MetaDataController
CI700.ci700_tag_cache.metadata_controller44.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller45	MetaDataController
CI700.ci700_tag_cache.metadata_controller45.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller46	MetaDataController
CI700.ci700_tag_cache.metadata_controller46.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller47	MetaDataController
CI700.ci700_tag_cache.metadata_controller47.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller48	MetaDataController

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller48.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller49	MetaDataController
CI700.ci700_tag_cache.metadata_controller49.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller5	MetaDataController
CI700.ci700_tag_cache.metadata_controller5.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller50	MetaDataController
CI700.ci700_tag_cache.metadata_controller50.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller51	MetaDataController
CI700.ci700_tag_cache.metadata_controller51.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller52	MetaDataController
CI700.ci700_tag_cache.metadata_controller52.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller53	MetaDataController
CI700.ci700_tag_cache.metadata_controller53.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller54	MetaDataController
CI700.ci700_tag_cache.metadata_controller54.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller55	MetaDataController
CI700.ci700_tag_cache.metadata_controller55.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller56	MetaDataController
CI700.ci700_tag_cache.metadata_controller56.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller57	MetaDataController
CI700.ci700_tag_cache.metadata_controller57.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller58	MetaDataController
CI700.ci700_tag_cache.metadata_controller58.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller59	MetaDataController
CI700.ci700_tag_cache.metadata_controller59.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller6	MetaDataController
CI700.ci700_tag_cache.metadata_controller6.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller60	MetaDataController
CI700.ci700_tag_cache.metadata_controller60.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller61	MetaDataController
CI700.ci700_tag_cache.metadata_controller61.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller62	MetaDataController
CI700.ci700_tag_cache.metadata_controller62.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller63	MetaDataController
CI700.ci700_tag_cache.metadata_controller63.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller64	MetaDataController
CI700.ci700_tag_cache.metadata_controller64.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller65	MetaDataController
CI700.ci700_tag_cache.metadata_controller65.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller66	MetaDataController
CI700.ci700_tag_cache.metadata_controller66.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller67	MetaDataController
CI700.ci700_tag_cache.metadata_controller67.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller68	MetaDataController
CI700.ci700_tag_cache.metadata_controller68.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller69	MetaDataController
CI700.ci700_tag_cache.metadata_controller69.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller7	MetaDataController
CI700.ci700_tag_cache.metadata_controller7.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller70	MetaDataController
CI700.ci700_tag_cache.metadata_controller70.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller71	MetaDataController
CI700.ci700_tag_cache.metadata_controller71.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller72	MetaDataController
CI700.ci700_tag_cache.metadata_controller72.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller73	MetaDataController
CI700.ci700_tag_cache.metadata_controller73.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller74	MetaDataController
CI700.ci700_tag_cache.metadata_controller74.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller75	MetaDataController
CI700.ci700_tag_cache.metadata_controller75.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller76	MetaDataController
CI700.ci700_tag_cache.metadata_controller76.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller77	MetaDataController
CI700.ci700_tag_cache.metadata_controller77.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller78	MetaDataController
CI700.ci700_tag_cache.metadata_controller78.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller79	MetaDataController
CI700.ci700_tag_cache.metadata_controller79.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller8	MetaDataController
CI700.ci700_tag_cache.metadata_controller8.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller80	MetaDataController
CI700.ci700_tag_cache.metadata_controller80.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller81	MetaDataController
CI700.ci700_tag_cache.metadata_controller81.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller82	MetaDataController
CI700.ci700_tag_cache.metadata_controller82.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller83	MetaDataController

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller83.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller84	MetaDataController
CI700.ci700_tag_cache.metadata_controller84.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller85	MetaDataController
CI700.ci700_tag_cache.metadata_controller85.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller86	MetaDataController
CI700.ci700_tag_cache.metadata_controller86.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller87	MetaDataController
CI700.ci700_tag_cache.metadata_controller87.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller88	MetaDataController
CI700.ci700_tag_cache.metadata_controller88.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller89	MetaDataController
CI700.ci700_tag_cache.metadata_controller89.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller9	MetaDataController
CI700.ci700_tag_cache.metadata_controller9.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller90	MetaDataController
CI700.ci700_tag_cache.metadata_controller90.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller91	MetaDataController
CI700.ci700_tag_cache.metadata_controller91.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller92	MetaDataController
CI700.ci700_tag_cache.metadata_controller92.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller93	MetaDataController
CI700.ci700_tag_cache.metadata_controller93.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller94	MetaDataController
CI700.ci700_tag_cache.metadata_controller94.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller95	MetaDataController
CI700.ci700_tag_cache.metadata_controller95.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller96	MetaDataController
CI700.ci700_tag_cache.metadata_controller96.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller97	MetaDataController
CI700.ci700_tag_cache.metadata_controller97.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller98	MetaDataController
CI700.ci700_tag_cache.metadata_controller98.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller99	MetaDataController
CI700.ci700_tag_cache.metadata_controller99.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.remapper0	PVBusMapper
CI700.ci700_tag_cache.remapper1	PVBusMapper
CI700.ci700_tag_cache.remapper10	PVBusMapper
CI700.ci700_tag_cache.remapper100	PVBusMapper



InstanceName	ComponentName
CI700.ci700_tag_cache.remapper101	PVBusMapper
CI700.ci700_tag_cache.remapper102	PVBusMapper
CI700.ci700_tag_cache.remapper103	PVBusMapper
CI700.ci700_tag_cache.remapper104	PVBusMapper
CI700.ci700_tag_cache.remapper105	PVBusMapper
CI700.ci700_tag_cache.remapper106	PVBusMapper
CI700.ci700_tag_cache.remapper107	PVBusMapper
CI700.ci700_tag_cache.remapper108	PVBusMapper
CI700.ci700_tag_cache.remapper109	PVBusMapper
CI700.ci700_tag_cache.remapper11	PVBusMapper
CI700.ci700_tag_cache.remapper110	PVBusMapper
CI700.ci700_tag_cache.remapper111	PVBusMapper
CI700.ci700_tag_cache.remapper112	PVBusMapper
CI700.ci700_tag_cache.remapper113	PVBusMapper
CI700.ci700_tag_cache.remapper114	PVBusMapper
CI700.ci700_tag_cache.remapper115	PVBusMapper
CI700.ci700_tag_cache.remapper116	PVBusMapper
CI700.ci700_tag_cache.remapper117	PVBusMapper
CI700.ci700_tag_cache.remapper118	PVBusMapper
CI700.ci700_tag_cache.remapper119	PVBusMapper
CI700.ci700_tag_cache.remapper12	PVBusMapper
CI700.ci700_tag_cache.remapper120	PVBusMapper
CI700.ci700_tag_cache.remapper121	PVBusMapper
CI700.ci700_tag_cache.remapper122	PVBusMapper
CI700.ci700_tag_cache.remapper123	PVBusMapper
CI700.ci700_tag_cache.remapper124	PVBusMapper
CI700.ci700_tag_cache.remapper125	PVBusMapper
CI700.ci700_tag_cache.remapper126	PVBusMapper
CI700.ci700_tag_cache.remapper127	PVBusMapper
CI700.ci700_tag_cache.remapper13	PVBusMapper
CI700.ci700_tag_cache.remapper14	PVBusMapper
CI700.ci700_tag_cache.remapper15	PVBusMapper
CI700.ci700_tag_cache.remapper16	PVBusMapper
CI700.ci700_tag_cache.remapper17	PVBusMapper
CI700.ci700_tag_cache.remapper18	PVBusMapper
CI700.ci700_tag_cache.remapper19	PVBusMapper
CI700.ci700_tag_cache.remapper2	PVBusMapper
CI700.ci700_tag_cache.remapper20	PVBusMapper
CI700.ci700_tag_cache.remapper21	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.remapper22	PVBusMapper
CI700.ci700_tag_cache.remapper23	PVBusMapper
CI700.ci700_tag_cache.remapper24	PVBusMapper
CI700.ci700_tag_cache.remapper25	PVBusMapper
CI700.ci700_tag_cache.remapper26	PVBusMapper
CI700.ci700_tag_cache.remapper27	PVBusMapper
CI700.ci700_tag_cache.remapper28	PVBusMapper
CI700.ci700_tag_cache.remapper29	PVBusMapper
CI700.ci700_tag_cache.remapper3	PVBusMapper
CI700.ci700_tag_cache.remapper30	PVBusMapper
CI700.ci700_tag_cache.remapper31	PVBusMapper
CI700.ci700_tag_cache.remapper32	PVBusMapper
CI700.ci700_tag_cache.remapper33	PVBusMapper
CI700.ci700_tag_cache.remapper34	PVBusMapper
CI700.ci700_tag_cache.remapper35	PVBusMapper
CI700.ci700_tag_cache.remapper36	PVBusMapper
CI700.ci700_tag_cache.remapper37	PVBusMapper
CI700.ci700_tag_cache.remapper38	PVBusMapper
CI700.ci700_tag_cache.remapper39	PVBusMapper
CI700.ci700_tag_cache.remapper4	PVBusMapper
CI700.ci700_tag_cache.remapper40	PVBusMapper
CI700.ci700_tag_cache.remapper41	PVBusMapper
CI700.ci700_tag_cache.remapper42	PVBusMapper
CI700.ci700_tag_cache.remapper43	PVBusMapper
CI700.ci700_tag_cache.remapper44	PVBusMapper
CI700.ci700_tag_cache.remapper45	PVBusMapper
CI700.ci700_tag_cache.remapper46	PVBusMapper
CI700.ci700_tag_cache.remapper47	PVBusMapper
CI700.ci700_tag_cache.remapper48	PVBusMapper
CI700.ci700_tag_cache.remapper49	PVBusMapper
CI700.ci700_tag_cache.remapper5	PVBusMapper
CI700.ci700_tag_cache.remapper50	PVBusMapper
CI700.ci700_tag_cache.remapper51	PVBusMapper
CI700.ci700_tag_cache.remapper52	PVBusMapper
CI700.ci700_tag_cache.remapper53	PVBusMapper
CI700.ci700_tag_cache.remapper54	PVBusMapper
CI700.ci700_tag_cache.remapper55	PVBusMapper
CI700.ci700_tag_cache.remapper56	PVBusMapper
CI700.ci700_tag_cache.remapper57	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.remapper58	PVBusMapper
CI700.ci700_tag_cache.remapper59	PVBusMapper
CI700.ci700_tag_cache.remapper6	PVBusMapper
CI700.ci700_tag_cache.remapper60	PVBusMapper
CI700.ci700_tag_cache.remapper61	PVBusMapper
CI700.ci700_tag_cache.remapper62	PVBusMapper
CI700.ci700_tag_cache.remapper63	PVBusMapper
CI700.ci700_tag_cache.remapper64	PVBusMapper
CI700.ci700_tag_cache.remapper65	PVBusMapper
CI700.ci700_tag_cache.remapper66	PVBusMapper
CI700.ci700_tag_cache.remapper67	PVBusMapper
CI700.ci700_tag_cache.remapper68	PVBusMapper
CI700.ci700_tag_cache.remapper69	PVBusMapper
CI700.ci700_tag_cache.remapper7	PVBusMapper
CI700.ci700_tag_cache.remapper70	PVBusMapper
CI700.ci700_tag_cache.remapper71	PVBusMapper
CI700.ci700_tag_cache.remapper72	PVBusMapper
CI700.ci700_tag_cache.remapper73	PVBusMapper
CI700.ci700_tag_cache.remapper74	PVBusMapper
CI700.ci700_tag_cache.remapper75	PVBusMapper
CI700.ci700_tag_cache.remapper76	PVBusMapper
CI700.ci700_tag_cache.remapper77	PVBusMapper
CI700.ci700_tag_cache.remapper78	PVBusMapper
CI700.ci700_tag_cache.remapper79	PVBusMapper
CI700.ci700_tag_cache.remapper8	PVBusMapper
CI700.ci700_tag_cache.remapper80	PVBusMapper
CI700.ci700_tag_cache.remapper81	PVBusMapper
CI700.ci700_tag_cache.remapper82	PVBusMapper
CI700.ci700_tag_cache.remapper83	PVBusMapper
CI700.ci700_tag_cache.remapper84	PVBusMapper
CI700.ci700_tag_cache.remapper85	PVBusMapper
CI700.ci700_tag_cache.remapper86	PVBusMapper
CI700.ci700_tag_cache.remapper87	PVBusMapper
CI700.ci700_tag_cache.remapper88	PVBusMapper
CI700.ci700_tag_cache.remapper89	PVBusMapper
CI700.ci700_tag_cache.remapper9	PVBusMapper
CI700.ci700_tag_cache.remapper90	PVBusMapper
CI700.ci700_tag_cache.remapper91	PVBusMapper
CI700.ci700_tag_cache.remapper92	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.remapper93	PVBusMapper
CI700.ci700_tag_cache.remapper94	PVBusMapper
CI700.ci700_tag_cache.remapper95	PVBusMapper
CI700.ci700_tag_cache.remapper96	PVBusMapper
CI700.ci700_tag_cache.remapper97	PVBusMapper
CI700.ci700_tag_cache.remapper98	PVBusMapper
CI700.ci700_tag_cache.remapper99	PVBusMapper
CI700.cmn600_cache	PVCache
CI700.cmn600_cache.upstream[0]	PVBusSlave
CI700.cmn600_cache.upstream[1]	PVBusSlave
CI700.cmn600_cache.upstream[2]	PVBusSlave
CI700.cmn600_cache.upstream[3]	PVBusSlave
CI700.hnf_exclusive_monitor	PVBusExclusiveMonitor
CI700.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CI700.hni_exclusive_monitor0	PVBusExclusiveMonitor
CI700.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CI700.ocm_decoder	PVBusMapper
CI700.ocm_exclusive_monitor	PVBusExclusiveMonitor
CI700.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CI700.snf_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-891: CI700 MTI instances**

InstanceName	ComponentName
CI700	CI700
CI700.bus_slave_ocm_NS	PVBusSlave
CI700.bus_slave_ocm_S	PVBusSlave
CI700.ci700_tag_cache	CMNTAGCACHECADI
CI700.ci700_tag_cache.metadata_controller0	MetaDataController
CI700.ci700_tag_cache.metadata_controller0.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller1	MetaDataController
CI700.ci700_tag_cache.metadata_controller1.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller10	MetaDataController
CI700.ci700_tag_cache.metadata_controller10.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller100	MetaDataController
CI700.ci700_tag_cache.metadata_controller100.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller101	MetaDataController
CI700.ci700_tag_cache.metadata_controller101.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller102	MetaDataController

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller102.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller103	MetadataController
CI700.ci700_tag_cache.metadata_controller103.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller104	MetadataController
CI700.ci700_tag_cache.metadata_controller104.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller105	MetadataController
CI700.ci700_tag_cache.metadata_controller105.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller106	MetadataController
CI700.ci700_tag_cache.metadata_controller106.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller107	MetadataController
CI700.ci700_tag_cache.metadata_controller107.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller108	MetadataController
CI700.ci700_tag_cache.metadata_controller108.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller109	MetadataController
CI700.ci700_tag_cache.metadata_controller109.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller11	MetadataController
CI700.ci700_tag_cache.metadata_controller11.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller110	MetadataController
CI700.ci700_tag_cache.metadata_controller110.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller111	MetadataController
CI700.ci700_tag_cache.metadata_controller111.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller112	MetadataController
CI700.ci700_tag_cache.metadata_controller112.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller113	MetadataController
CI700.ci700_tag_cache.metadata_controller113.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller114	MetadataController
CI700.ci700_tag_cache.metadata_controller114.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller115	MetadataController
CI700.ci700_tag_cache.metadata_controller115.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller116	MetadataController
CI700.ci700_tag_cache.metadata_controller116.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller117	MetadataController
CI700.ci700_tag_cache.metadata_controller117.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller118	MetadataController
CI700.ci700_tag_cache.metadata_controller118.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller119	MetadataController
CI700.ci700_tag_cache.metadata_controller119.MetadataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller12	MetadataController
CI700.ci700_tag_cache.metadata_controller12.MetadataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller120	MetaDataController
CI700.ci700_tag_cache.metadata_controller120.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller121	MetaDataController
CI700.ci700_tag_cache.metadata_controller121.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller122	MetaDataController
CI700.ci700_tag_cache.metadata_controller122.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller123	MetaDataController
CI700.ci700_tag_cache.metadata_controller123.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller124	MetaDataController
CI700.ci700_tag_cache.metadata_controller124.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller125	MetaDataController
CI700.ci700_tag_cache.metadata_controller125.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller126	MetaDataController
CI700.ci700_tag_cache.metadata_controller126.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller127	MetaDataController
CI700.ci700_tag_cache.metadata_controller127.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller13	MetaDataController
CI700.ci700_tag_cache.metadata_controller13.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller14	MetaDataController
CI700.ci700_tag_cache.metadata_controller14.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller15	MetaDataController
CI700.ci700_tag_cache.metadata_controller15.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller16	MetaDataController
CI700.ci700_tag_cache.metadata_controller16.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller17	MetaDataController
CI700.ci700_tag_cache.metadata_controller17.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller18	MetaDataController
CI700.ci700_tag_cache.metadata_controller18.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller19	MetaDataController
CI700.ci700_tag_cache.metadata_controller19.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller2	MetaDataController
CI700.ci700_tag_cache.metadata_controller2.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller20	MetaDataController
CI700.ci700_tag_cache.metadata_controller20.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller21	MetaDataController
CI700.ci700_tag_cache.metadata_controller21.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller22	MetaDataController
CI700.ci700_tag_cache.metadata_controller22.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller23	MetaDataController

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller23.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller24	MetaDataController
CI700.ci700_tag_cache.metadata_controller24.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller25	MetaDataController
CI700.ci700_tag_cache.metadata_controller25.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller26	MetaDataController
CI700.ci700_tag_cache.metadata_controller26.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller27	MetaDataController
CI700.ci700_tag_cache.metadata_controller27.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller28	MetaDataController
CI700.ci700_tag_cache.metadata_controller28.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller29	MetaDataController
CI700.ci700_tag_cache.metadata_controller29.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller3	MetaDataController
CI700.ci700_tag_cache.metadata_controller3.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller30	MetaDataController
CI700.ci700_tag_cache.metadata_controller30.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller31	MetaDataController
CI700.ci700_tag_cache.metadata_controller31.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller32	MetaDataController
CI700.ci700_tag_cache.metadata_controller32.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller33	MetaDataController
CI700.ci700_tag_cache.metadata_controller33.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller34	MetaDataController
CI700.ci700_tag_cache.metadata_controller34.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller35	MetaDataController
CI700.ci700_tag_cache.metadata_controller35.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller36	MetaDataController
CI700.ci700_tag_cache.metadata_controller36.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller37	MetaDataController
CI700.ci700_tag_cache.metadata_controller37.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller38	MetaDataController
CI700.ci700_tag_cache.metadata_controller38.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller39	MetaDataController
CI700.ci700_tag_cache.metadata_controller39.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller4	MetaDataController
CI700.ci700_tag_cache.metadata_controller4.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller40	MetaDataController
CI700.ci700_tag_cache.metadata_controller40.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller41	MetaDataController
CI700.ci700_tag_cache.metadata_controller41.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller42	MetaDataController
CI700.ci700_tag_cache.metadata_controller42.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller43	MetaDataController
CI700.ci700_tag_cache.metadata_controller43.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller44	MetaDataController
CI700.ci700_tag_cache.metadata_controller44.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller45	MetaDataController
CI700.ci700_tag_cache.metadata_controller45.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller46	MetaDataController
CI700.ci700_tag_cache.metadata_controller46.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller47	MetaDataController
CI700.ci700_tag_cache.metadata_controller47.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller48	MetaDataController
CI700.ci700_tag_cache.metadata_controller48.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller49	MetaDataController
CI700.ci700_tag_cache.metadata_controller49.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller5	MetaDataController
CI700.ci700_tag_cache.metadata_controller5.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller50	MetaDataController
CI700.ci700_tag_cache.metadata_controller50.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller51	MetaDataController
CI700.ci700_tag_cache.metadata_controller51.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller52	MetaDataController
CI700.ci700_tag_cache.metadata_controller52.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller53	MetaDataController
CI700.ci700_tag_cache.metadata_controller53.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller54	MetaDataController
CI700.ci700_tag_cache.metadata_controller54.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller55	MetaDataController
CI700.ci700_tag_cache.metadata_controller55.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller56	MetaDataController
CI700.ci700_tag_cache.metadata_controller56.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller57	MetaDataController
CI700.ci700_tag_cache.metadata_controller57.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller58	MetaDataController
CI700.ci700_tag_cache.metadata_controller58.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller59	MetaDataController



InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller59.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller6	MetaDataController
CI700.ci700_tag_cache.metadata_controller6.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller60	MetaDataController
CI700.ci700_tag_cache.metadata_controller60.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller61	MetaDataController
CI700.ci700_tag_cache.metadata_controller61.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller62	MetaDataController
CI700.ci700_tag_cache.metadata_controller62.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller63	MetaDataController
CI700.ci700_tag_cache.metadata_controller63.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller64	MetaDataController
CI700.ci700_tag_cache.metadata_controller64.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller65	MetaDataController
CI700.ci700_tag_cache.metadata_controller65.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller66	MetaDataController
CI700.ci700_tag_cache.metadata_controller66.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller67	MetaDataController
CI700.ci700_tag_cache.metadata_controller67.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller68	MetaDataController
CI700.ci700_tag_cache.metadata_controller68.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller69	MetaDataController
CI700.ci700_tag_cache.metadata_controller69.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller7	MetaDataController
CI700.ci700_tag_cache.metadata_controller7.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller70	MetaDataController
CI700.ci700_tag_cache.metadata_controller70.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller71	MetaDataController
CI700.ci700_tag_cache.metadata_controller71.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller72	MetaDataController
CI700.ci700_tag_cache.metadata_controller72.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller73	MetaDataController
CI700.ci700_tag_cache.metadata_controller73.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller74	MetaDataController
CI700.ci700_tag_cache.metadata_controller74.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller75	MetaDataController
CI700.ci700_tag_cache.metadata_controller75.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller76	MetaDataController
CI700.ci700_tag_cache.metadata_controller76.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller77	MetaDataController
CI700.ci700_tag_cache.metadata_controller77.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller78	MetaDataController
CI700.ci700_tag_cache.metadata_controller78.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller79	MetaDataController
CI700.ci700_tag_cache.metadata_controller79.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller8	MetaDataController
CI700.ci700_tag_cache.metadata_controller8.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller80	MetaDataController
CI700.ci700_tag_cache.metadata_controller80.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller81	MetaDataController
CI700.ci700_tag_cache.metadata_controller81.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller82	MetaDataController
CI700.ci700_tag_cache.metadata_controller82.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller83	MetaDataController
CI700.ci700_tag_cache.metadata_controller83.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller84	MetaDataController
CI700.ci700_tag_cache.metadata_controller84.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller85	MetaDataController
CI700.ci700_tag_cache.metadata_controller85.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller86	MetaDataController
CI700.ci700_tag_cache.metadata_controller86.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller87	MetaDataController
CI700.ci700_tag_cache.metadata_controller87.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller88	MetaDataController
CI700.ci700_tag_cache.metadata_controller88.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller89	MetaDataController
CI700.ci700_tag_cache.metadata_controller89.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller9	MetaDataController
CI700.ci700_tag_cache.metadata_controller9.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller90	MetaDataController
CI700.ci700_tag_cache.metadata_controller90.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller91	MetaDataController
CI700.ci700_tag_cache.metadata_controller91.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller92	MetaDataController
CI700.ci700_tag_cache.metadata_controller92.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller93	MetaDataController
CI700.ci700_tag_cache.metadata_controller93.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller94	MetaDataController

InstanceName	ComponentName
CI700.ci700_tag_cache.metadata_controller94.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller95	MetaDataController
CI700.ci700_tag_cache.metadata_controller95.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller96	MetaDataController
CI700.ci700_tag_cache.metadata_controller96.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller97	MetaDataController
CI700.ci700_tag_cache.metadata_controller97.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller98	MetaDataController
CI700.ci700_tag_cache.metadata_controller98.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.metadata_controller99	MetaDataController
CI700.ci700_tag_cache.metadata_controller99.MetaDataMapper	PVBusMapper
CI700.ci700_tag_cache.remapper0	PVBusMapper
CI700.ci700_tag_cache.remapper1	PVBusMapper
CI700.ci700_tag_cache.remapper10	PVBusMapper
CI700.ci700_tag_cache.remapper100	PVBusMapper
CI700.ci700_tag_cache.remapper101	PVBusMapper
CI700.ci700_tag_cache.remapper102	PVBusMapper
CI700.ci700_tag_cache.remapper103	PVBusMapper
CI700.ci700_tag_cache.remapper104	PVBusMapper
CI700.ci700_tag_cache.remapper105	PVBusMapper
CI700.ci700_tag_cache.remapper106	PVBusMapper
CI700.ci700_tag_cache.remapper107	PVBusMapper
CI700.ci700_tag_cache.remapper108	PVBusMapper
CI700.ci700_tag_cache.remapper109	PVBusMapper
CI700.ci700_tag_cache.remapper11	PVBusMapper
CI700.ci700_tag_cache.remapper110	PVBusMapper
CI700.ci700_tag_cache.remapper111	PVBusMapper
CI700.ci700_tag_cache.remapper112	PVBusMapper
CI700.ci700_tag_cache.remapper113	PVBusMapper
CI700.ci700_tag_cache.remapper114	PVBusMapper
CI700.ci700_tag_cache.remapper115	PVBusMapper
CI700.ci700_tag_cache.remapper116	PVBusMapper
CI700.ci700_tag_cache.remapper117	PVBusMapper
CI700.ci700_tag_cache.remapper118	PVBusMapper
CI700.ci700_tag_cache.remapper119	PVBusMapper
CI700.ci700_tag_cache.remapper12	PVBusMapper
CI700.ci700_tag_cache.remapper120	PVBusMapper
CI700.ci700_tag_cache.remapper121	PVBusMapper
CI700.ci700_tag_cache.remapper122	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.remapper123	PVBusMapper
CI700.ci700_tag_cache.remapper124	PVBusMapper
CI700.ci700_tag_cache.remapper125	PVBusMapper
CI700.ci700_tag_cache.remapper126	PVBusMapper
CI700.ci700_tag_cache.remapper127	PVBusMapper
CI700.ci700_tag_cache.remapper13	PVBusMapper
CI700.ci700_tag_cache.remapper14	PVBusMapper
CI700.ci700_tag_cache.remapper15	PVBusMapper
CI700.ci700_tag_cache.remapper16	PVBusMapper
CI700.ci700_tag_cache.remapper17	PVBusMapper
CI700.ci700_tag_cache.remapper18	PVBusMapper
CI700.ci700_tag_cache.remapper19	PVBusMapper
CI700.ci700_tag_cache.remapper2	PVBusMapper
CI700.ci700_tag_cache.remapper20	PVBusMapper
CI700.ci700_tag_cache.remapper21	PVBusMapper
CI700.ci700_tag_cache.remapper22	PVBusMapper
CI700.ci700_tag_cache.remapper23	PVBusMapper
CI700.ci700_tag_cache.remapper24	PVBusMapper
CI700.ci700_tag_cache.remapper25	PVBusMapper
CI700.ci700_tag_cache.remapper26	PVBusMapper
CI700.ci700_tag_cache.remapper27	PVBusMapper
CI700.ci700_tag_cache.remapper28	PVBusMapper
CI700.ci700_tag_cache.remapper29	PVBusMapper
CI700.ci700_tag_cache.remapper3	PVBusMapper
CI700.ci700_tag_cache.remapper30	PVBusMapper
CI700.ci700_tag_cache.remapper31	PVBusMapper
CI700.ci700_tag_cache.remapper32	PVBusMapper
CI700.ci700_tag_cache.remapper33	PVBusMapper
CI700.ci700_tag_cache.remapper34	PVBusMapper
CI700.ci700_tag_cache.remapper35	PVBusMapper
CI700.ci700_tag_cache.remapper36	PVBusMapper
CI700.ci700_tag_cache.remapper37	PVBusMapper
CI700.ci700_tag_cache.remapper38	PVBusMapper
CI700.ci700_tag_cache.remapper39	PVBusMapper
CI700.ci700_tag_cache.remapper4	PVBusMapper
CI700.ci700_tag_cache.remapper40	PVBusMapper
CI700.ci700_tag_cache.remapper41	PVBusMapper
CI700.ci700_tag_cache.remapper42	PVBusMapper
CI700.ci700_tag_cache.remapper43	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.remapper44	PVBusMapper
CI700.ci700_tag_cache.remapper45	PVBusMapper
CI700.ci700_tag_cache.remapper46	PVBusMapper
CI700.ci700_tag_cache.remapper47	PVBusMapper
CI700.ci700_tag_cache.remapper48	PVBusMapper
CI700.ci700_tag_cache.remapper49	PVBusMapper
CI700.ci700_tag_cache.remapper5	PVBusMapper
CI700.ci700_tag_cache.remapper50	PVBusMapper
CI700.ci700_tag_cache.remapper51	PVBusMapper
CI700.ci700_tag_cache.remapper52	PVBusMapper
CI700.ci700_tag_cache.remapper53	PVBusMapper
CI700.ci700_tag_cache.remapper54	PVBusMapper
CI700.ci700_tag_cache.remapper55	PVBusMapper
CI700.ci700_tag_cache.remapper56	PVBusMapper
CI700.ci700_tag_cache.remapper57	PVBusMapper
CI700.ci700_tag_cache.remapper58	PVBusMapper
CI700.ci700_tag_cache.remapper59	PVBusMapper
CI700.ci700_tag_cache.remapper6	PVBusMapper
CI700.ci700_tag_cache.remapper60	PVBusMapper
CI700.ci700_tag_cache.remapper61	PVBusMapper
CI700.ci700_tag_cache.remapper62	PVBusMapper
CI700.ci700_tag_cache.remapper63	PVBusMapper
CI700.ci700_tag_cache.remapper64	PVBusMapper
CI700.ci700_tag_cache.remapper65	PVBusMapper
CI700.ci700_tag_cache.remapper66	PVBusMapper
CI700.ci700_tag_cache.remapper67	PVBusMapper
CI700.ci700_tag_cache.remapper68	PVBusMapper
CI700.ci700_tag_cache.remapper69	PVBusMapper
CI700.ci700_tag_cache.remapper7	PVBusMapper
CI700.ci700_tag_cache.remapper70	PVBusMapper
CI700.ci700_tag_cache.remapper71	PVBusMapper
CI700.ci700_tag_cache.remapper72	PVBusMapper
CI700.ci700_tag_cache.remapper73	PVBusMapper
CI700.ci700_tag_cache.remapper74	PVBusMapper
CI700.ci700_tag_cache.remapper75	PVBusMapper
CI700.ci700_tag_cache.remapper76	PVBusMapper
CI700.ci700_tag_cache.remapper77	PVBusMapper
CI700.ci700_tag_cache.remapper78	PVBusMapper
CI700.ci700_tag_cache.remapper79	PVBusMapper

InstanceName	ComponentName
CI700.ci700_tag_cache.remapper8	PVBusMapper
CI700.ci700_tag_cache.remapper80	PVBusMapper
CI700.ci700_tag_cache.remapper81	PVBusMapper
CI700.ci700_tag_cache.remapper82	PVBusMapper
CI700.ci700_tag_cache.remapper83	PVBusMapper
CI700.ci700_tag_cache.remapper84	PVBusMapper
CI700.ci700_tag_cache.remapper85	PVBusMapper
CI700.ci700_tag_cache.remapper86	PVBusMapper
CI700.ci700_tag_cache.remapper87	PVBusMapper
CI700.ci700_tag_cache.remapper88	PVBusMapper
CI700.ci700_tag_cache.remapper89	PVBusMapper
CI700.ci700_tag_cache.remapper9	PVBusMapper
CI700.ci700_tag_cache.remapper90	PVBusMapper
CI700.ci700_tag_cache.remapper91	PVBusMapper
CI700.ci700_tag_cache.remapper92	PVBusMapper
CI700.ci700_tag_cache.remapper93	PVBusMapper
CI700.ci700_tag_cache.remapper94	PVBusMapper
CI700.ci700_tag_cache.remapper95	PVBusMapper
CI700.ci700_tag_cache.remapper96	PVBusMapper
CI700.ci700_tag_cache.remapper97	PVBusMapper
CI700.ci700_tag_cache.remapper98	PVBusMapper
CI700.ci700_tag_cache.remapper99	PVBusMapper
CI700.cmn600_cache	CMN600Cache
CI700.cmn600_cache.upstream[0]	PVBusSlave
CI700.cmn600_cache.upstream[1]	PVBusSlave
CI700.cmn600_cache.upstream[2]	PVBusSlave
CI700.cmn600_cache.upstream[3]	PVBusSlave
CI700.hnf_exclusive_monitor	PVBusExclusiveMonitor
CI700.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CI700.hni_exclusive_monitor0	PVBusExclusiveMonitor
CI700.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CI700.ocm_decoder	PVBusMapper
CI700.ocm_exclusive_monitor	PVBusExclusiveMonitor
CI700.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CI700.snf_mapper	PVBusMapper

## Ports for CI700

Table 3-892: Ports

Name	Protocol	Type	Description
event_clusters	Signal	Peer	CPU event communication signal from the clusters.
pvbuss_m_hni[4]	PVBus	Master	HNI downstream ports.
pvbuss_m_snf[8]	PVBus	Master	SNF downstream port.
pvbuss_s_apb	PVBus	Slave	APB interface port.
pvbuss_s_rnf[8]	PVBus	Slave	RNF upstream ports.
pvbuss_s_rni[24]	PVBus	Slave	RNI upstream ports.
reset_in	Signal	Slave	Reset signal.
rnf_sci_s[8]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in[8]	Signal	Slave	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s[24]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in[24]	Signal	Slave	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

## Parameters for CI700

**acchannelen\_rnf****Type**

string

**Default value**

"0"

**Note**

DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each rnf port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`

**acchannelen\_rni****Type**

string

**Default value**

"0"

**Note**

DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`

### **bypass\_tag\_cache**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

If true, CI700 will bypass the tag cache component which provides the MTE support.

### **cache\_state\_modelled**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Model the cache state.

### **ci700\_tag\_cache.metadata\_controller.init\_value**

#### **Type**

int

#### **Default value**

0xd

#### **Description**

Initialize metadata memory with this value. If one of `init_values_json` or `init_values_json_file` is specified, this value applies only to any metadata not specified in the JSON.

### **ci700\_tag\_cache.metadata\_controller.init\_values\_json**

#### **Type**

string

#### **Default value**

""



**Description**

A JSON value describing initial metadata values. Mutually exclusive with `init_values_json_file`. The format is as follows: `{"regions": [{ "begin": 0x0, "end": 0x10000, "mte_tag": 0xa }, {"begin": 0x20000, "end": 0x50000, "mte_tag": 0xc }]}`.

**`ci700_tag_cache.metadata_controller.init_values_json_file`****Type**

string

**Default value**

""

**Description**

Path to a JSON file with initial metadata values. Mutually exclusive with `init_values_json`. The format is as follows: `{"regions": [{ "begin": 0x0, "end": 0x10000, "mte_tag": 0xa }, {"begin": 0x20000, "end": 0x50000, "mte_tag": 0xc }]}`.

**`ci700_tag_cache.metadata_controller.is_enabled`****Type**

bool

**Default value**

0x0

**Description**

If false, disables the MetaData controller functionality, and makes the component invisible to passing transactions.

**`ci700_tag_cache.metadata_controller.mte_tag_carveout_json`****Type**

string

**Default value**

""

**Description**

JSON string that specifies the PA range of the tag carveout regions and the beginning of the PA range for which they provide tag storage. If `pa_regions_with_metadata_storage` defines which regions can have metadata, the tag carveout regions cannot overlap them, and each tagged region must be entirely covered by one of them. The block size must be  $\geq 64$  bytes and a power of 2, defaulting to 4KiB. The maximum block size supported is 4KiB. The carveout region size must be  $\geq 4$ KiB and a power of 2, and determines the size of the corresponding tagged region. `{ "regions": [ { "begin": 0x0, "tag_carveout_region": [0xfffff00000, 0xfffff00fff] }, { "begin": 0x20000, "tag_carveout_region": [0xfffff01000, 0xfffff01fff], "block_size": 0x100 }, { "begin": 0x100000, "tag_carveout_region": [0xfffff08000, 0xfffff0Bfff], "block_size": 0x2000 } ] }`.

**ci700\_tag\_cache.metadata\_controller.mte\_tag\_carveout\_json\_file****Type**

string

**Default value**

""

**Description**

Path to a JSON file that specifies the PA range of the tag carveout regions with the same format as mte\_tag\_carveout\_json. Only one of mte\_tag\_carveout\_json and mte\_tag\_carveout\_json\_file can be used. .

**ci700\_tag\_cache.metadata\_controller.mte\_tag\_carveout\_tag\_order****Type**

string

**Default value**

little-endian

**Description**

Order of the tags within the MTE tag carveout blocks. This can be little-endian (same order as the corresponding tagged data) or big-endian (reverse order). The parameter accepts both '-' and '\_', so 'little-endian', 'big-endian', 'little\_endian' and 'big\_endian' are all valid. .

**ci700\_tag\_cache.metadata\_controller.pa\_regions\_with\_metadata\_storage****Type**

string

**Default value**

""

**Description**

Specify the address region where the metadata storage is available for each PAS in a JSON format. If the PAS does not have a region specified, the PAS has metadata storage for all of the space. The regions are defined by begin and end\_incl addresses. Example: { "ns": [0xa0000000, 0xa0000fff], "s": [0xb0000000, 0xb0000fff], "rl": [0xc0000000, 0xc0000fff], "rt": [0xd0000000, 0xd0000fff] } ns: non-secure, s: secure, rl: realm, rt: root.

**debug\_force\_snoop****Type**

bool

**Default value**

false

The CI700 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

**Note**

This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

### **`enable_logger`**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Enable PVBUSLoggers for the downstream ports in the CMN model.

### **`enable_rnsam_to_hnf_wider_hash`**

#### **Type**

bool

#### **Default value**

false

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

### **`force_rnsam_internal`**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

Force all RNSAMs to be internal independently of the mesh topology.

**hnf\_mpam\_idr\_override****Type**

uint64\_t

**Default value**

0

Set to override hnf\_mpam\_idr[31:24] value. Bit[28] is Reserved and is ignored.

**mesh\_config\_file****Type**

string

**Default value**

""

**Description**

Name of a file containing mesh placement of CI700 components.

**periphbase****Type**

uint64\_t

**Default value**

0x20000000

Value for PERIPHBASE. Bits [27:0] are treated 0

**print\_cmn\_ccix\_config****Type**

bool

**Default value**

0x0

**Description**

Print information about the CCIX configuration.

**print\_cmn\_config****Type**

bool

**Default value**

0x0

**Description**

Print the mesh topology and children pointers acquired from the YML file.

**revision****Type**

string

**Default value**

"r0p0"

Component revision.

Currently supports r2p0, r1p0, r0p0.

**rnf\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

**rni\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

**show\_banner****Type**

uint64\_t

**Default value**

2

Show component banner:

**0**

supress entire banner

**1**

suppress config file

**2+**

show full banner

**skip\_cmn\_config\_check****Type**

bool

**Default value**

0x0

**Description**

Skip any topology configuration checks. The maximum number of devices per type not verified.

**use\_yaml\_periphbase****Type**

bool

**Default value**

false

Use yaml param CFGM\_PERIPHBASE\_PARAM to specify periphbase address.

If false, model parameter periphbase will be used.

**yaml\_has\_node\_addresses****Type**

bool

**Default value**

0x0

**Description**

Does the top-level YAML file describe node-addresses ?.

### 3.10.14 CMN600

CMN600 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-893: IP revisions support**

Revision	Quality level
r1p1	Full support
r3p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About CMN600

- Major IP revisions (rX) are modeled and are controlled by the `revision` parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of the CHI nodes. Set it to the name of the yml configuration file emitted by Socrates. You must use version r1p4-01rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.

The model supports the following features:

- `rnf`, `rni/rnd`, `hni`, and `snf/sbsx` interface ports. The mapping between the port number and `nodeId` is based on the `nodeId` index. For example, RNF2 controls `pvbuss_s_rnf[2]`. Its index is specified in the `node_info` register as `logical_id`. Similar behavior can be expected for HN-I.

If both RN-D and RN-I nodes are present, then all starting `rni` ports are mapped to RN-D nodes and then the RN-I nodes. For example, for CMN600 with two RN-D nodes, one RN-I node, and given each RN-I or RN-D node controls three interface ports, `pvbuss_s_rni[0-2]` maps to RND0, `pvbuss_s_rni[3-5]` maps to RND1 and `pvbuss_s_rni[6-8]` maps to RNI0.

Similarly, SN-F and SBSX nodes are mapped to `pvbuss_m_snf[]` ports where starting ports are mapped to SN-F and then SBSX nodes.

- Mapping HN\*\_SLC\_SIZE\_PARAM values to cache sizes:

-8

OKB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16



**Note**

This value is not supported by the model.

**-2**128KB where `HN*_SLC_NUM_WAYS_PARAM=16`**-1**256KB where `HN*_SLC_NUM_WAYS_PARAM=16`**0**512KB where `HN*_SLC_NUM_WAYS_PARAM=16`**1**1MB where `HN*_SLC_NUM_WAYS_PARAM=16`**2**2MB where `HN*_SLC_NUM_WAYS_PARAM=16`**3**3MB where `HN*_SLC_NUM_WAYS_PARAM=12`**3**4MB where `HN*_SLC_NUM_WAYS_PARAM=16`

- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
  - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.
  - When `HNF_ABF_PR.abf_mode` is Reserved and `HNF_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
  - Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
  - The optional interrupt `INTREQPPU` is not supported.

## Model limitations

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
  - Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.



- Snoop filtering is not supported.
- Prefetch Target operations are not supported.
- The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.
- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-lite port have not been tested.
- RNSAMs external to the mesh network are not functionally supported.



Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31]) of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address [MAX:12] instead of the actual address [MAX:6], due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address [MAX:6], but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.
- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.
- SLC/LCC partitioning dynamic allocation is not supported.
- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mxp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0
0x00000002000 - 0x00000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- Transactions to addresses unused by Device registers in the Configuration Register Space are always routed to the Configuration Space even when the SAM is configured to use those addresses.

- GIC communication over A4S ports is not supported.
- CAL (Component Aggregation Layer) r2 and r3 features are supported. These features have limited testing.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.
- For Power State Transition related flush/disable of the SLC due to HNF\_PPU\_PWPR register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.
- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.
- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- In revision r3p0, for an HTG/SCG, the model does not consider the secondary region if the primary region is not valid.
- There is no support for RAS.
- The following limitations apply to System Cache Groups and Hash Target Groups:
  - A mix of local and remote HN-F targets is not supported.
  - CCG/CXRH target IDs in the HN target ID table are not supported (`sys_cache_grp_hn_nodeid_reg`).
  - The `hashed_target_grp_hnp_nodeid_reg` which is used for CCG/HN-P target IDs is not supported.
  - System/Hash Target Groups only support HN-Fs.
  - AXID hashing across HN-P/CCGs is not supported.
- HN-F SAM:
  - Address masking in default hash regions in HN-F SAM is not supported.
  - Hashing in default hash regions in HN-F SAM is not supported. However, trace `SNF_HASHING_Target_SNF` displays the target SNF as if 3SN/6SN hashing were enabled.



The transaction is always routed to the SN0 nodeid programmed in the SAM\_CONTROL register.

- Re-programming regions in HN-F SAM is not tested.
- Hashing across CCGs in HN-F SAM is not supported.
- For RN-D nodes, when software writes SYSCOREQ, DVM propagation gets enabled but SYSCOACK is not set.

### About the `debug_force_snoop` parameter

The CMN interconnect normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream `rnf` ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



This parameter is for debug purposes only. Do not use it instead of correctly configuring CMN.

---

### Iris and MTI instances for CMN600

This model has the following Iris instances:

**Table 3-894: CMN600 Iris instances**

InstanceName	ComponentName
CMN600	CMN600
CMN600.bus_slave_ocm_NS	PVBusSlave
CMN600.bus_slave_ocm_S	PVBusSlave
CMN600.cmn600_cache	CMN600Cache
CMN600.cmn600_cache.upstream[0]	PVBusSlave
CMN600.cmn600_cache.upstream[10]	PVBusSlave
CMN600.cmn600_cache.upstream[11]	PVBusSlave
CMN600.cmn600_cache.upstream[12]	PVBusSlave
CMN600.cmn600_cache.upstream[13]	PVBusSlave
CMN600.cmn600_cache.upstream[14]	PVBusSlave
CMN600.cmn600_cache.upstream[15]	PVBusSlave
CMN600.cmn600_cache.upstream[16]	PVBusSlave
CMN600.cmn600_cache.upstream[17]	PVBusSlave
CMN600.cmn600_cache.upstream[1]	PVBusSlave
CMN600.cmn600_cache.upstream[2]	PVBusSlave
CMN600.cmn600_cache.upstream[3]	PVBusSlave
CMN600.cmn600_cache.upstream[4]	PVBusSlave
CMN600.cmn600_cache.upstream[5]	PVBusSlave
CMN600.cmn600_cache.upstream[6]	PVBusSlave
CMN600.cmn600_cache.upstream[7]	PVBusSlave
CMN600.cmn600_cache.upstream[8]	PVBusSlave
CMN600.cmn600_cache.upstream[9]	PVBusSlave
CMN600.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN600.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN600.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN600.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN600.ocm_decoder	PVBusMapper
CMN600.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN600.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN600.snf_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-895: CMN600 MTI instances**

InstanceName	ComponentName
CMN600	CMN600
CMN600.bus_slave_ocm_NS	PVBusSlave
CMN600.bus_slave_ocm_S	PVBusSlave
CMN600.cmn600_cache	CMN600Cache

InstanceName	ComponentName
CMN600.cmn600_cache.upstream[0]	PVBusSlave
CMN600.cmn600_cache.upstream[10]	PVBusSlave
CMN600.cmn600_cache.upstream[11]	PVBusSlave
CMN600.cmn600_cache.upstream[12]	PVBusSlave
CMN600.cmn600_cache.upstream[13]	PVBusSlave
CMN600.cmn600_cache.upstream[14]	PVBusSlave
CMN600.cmn600_cache.upstream[15]	PVBusSlave
CMN600.cmn600_cache.upstream[16]	PVBusSlave
CMN600.cmn600_cache.upstream[17]	PVBusSlave
CMN600.cmn600_cache.upstream[1]	PVBusSlave
CMN600.cmn600_cache.upstream[2]	PVBusSlave
CMN600.cmn600_cache.upstream[3]	PVBusSlave
CMN600.cmn600_cache.upstream[4]	PVBusSlave
CMN600.cmn600_cache.upstream[5]	PVBusSlave
CMN600.cmn600_cache.upstream[6]	PVBusSlave
CMN600.cmn600_cache.upstream[7]	PVBusSlave
CMN600.cmn600_cache.upstream[8]	PVBusSlave
CMN600.cmn600_cache.upstream[9]	PVBusSlave
CMN600.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN600.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN600.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN600.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN600.ocm_decoder	PVBusMapper
CMN600.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN600.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN600.snf_mapper	PVBusMapper

## Ports for CMN600

**Table 3-896: Ports**

Name	Protocol	Type	Description
event_clusters	Signal	Peer	CPU event communication signal from the clusters.
event_downstream_link_signal[6]	Signal	Master	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.
event_upstream_link_signal[6]	Signal	Slave	Event from the Hub towards the CMN
pvbus_m_cml	PVBus	Master	CML downstream ports
pvbus_m_cml_cfg	PVBus	Master	CML downstream hub configuration port
pvbus_m_hni[8]	PVBus	Master	HNI downstream ports.
pvbus_m_snf[16]	PVBus	Master	SNF downstream port.

Name	Protocol	Type	Description
pvbuss_s_cml	PVBus	Slave	CML upstream ports
pvbuss_s_rnf[64]	PVBus	Slave	RNF upstream ports.
pvbuss_s_rni[96]	PVBus	Slave	RNI upstream ports.
reset_in	Signal	Slave	Reset signal.
rnf_sci_s[64]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in[64]	Signal	Slave	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s[96]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in[96]	Signal	Slave	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

## Parameters for CMN600

### **acchannelen\_rnf**

#### Type

string

#### Default value

"0"



DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each rnf port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`

### **acchannelen\_rni**

#### Type

string

#### Default value

"0"



DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `fff`

### **cache\_state\_modelled**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Model the cache state.

### **debug\_force\_snoop**

#### **Type**

bool

#### **Default value**

false

The CMN600 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.



This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

### **disable\_CML\_port**

#### **Type**

bool

#### **Default value**

0x0

**Description**

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enable PVBusLoggers for the downstream ports in the CMN model.

**enable\_rnsam\_to\_hnf\_wider\_hash****Type**

bool

**Default value**

false

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

**force\_rnsam\_internal****Type**

bool

**Default value**

0x1

**Description**

Force all RNSAMs to be internal independently of the mesh topology.

**mesh\_config\_file****Type**

string

**Default value**

""

**Description**

Name of a file containing mesh placement of CMN600 components.



**periphbase****Type**

uint64\_t

**Default value**

0x20000000

Value for PERIPHBASE. Bits [25:0] are treated 0

**print\_cmn\_ccix\_config****Type**

bool

**Default value**

0x0

**Description**

Print information about the CCIX configuration.

**print\_cmn\_config****Type**

bool

**Default value**

0x0

**Description**

Print the mesh topology and children pointers acquired from the YAML file.

**revision****Type**

string

**Default value**

"r1p1"

Component revision.

Currently supports r1p1, r3p0.

**rnf\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

### **rni\_sci\_enable**

#### **Type**

string

#### **Default value**

"0x0"

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

### **show\_banner**

#### **Type**

uint64\_t

#### **Default value**

2

Show component banner:

**0**

suppress entire banner

**1**

suppress config file

**2+**

show full banner

### **skip\_cmn\_config\_check**

#### **Type**

bool

**Default value**

0x0

**Description**

Skip any topology configuration checks. The maximum number of devices per type not verified.

**use\_yaml\_periphbase**

**Type**

bool

**Default value**

false

Use yaml param CFGM\_PERIPHBASE\_PARAM to specify periphbase address.

If false, model parameter periphbase will be used.

3.10.15 CMN600AE

CMN600AE Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-897: IP revisions support

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About CMN600AE

- Major IP revisions (rX) are modeled and are controlled by the revision parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the periph\_id2 register.
- To configure the model, you must have installed Arm Socrates. The mesh\_config\_file parameter defines the mesh placement of the CHI nodes. Set it to the name of the yaml configuration file emitted by Socrates. You must use version r1p4-01rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.

The CMN600AE model supports the following features:

- rnf, rni/rnd, hni, and snf/sbsx interface ports. The mapping between the port number and NodeId is based on the NodeId index. For example, RNF2 controls pvbuss\_rnf[2]. Its index is specified in the node\_info register as logical\_id. Similar behavior can be expected for HN-I.

If both RN-D and RN-I nodes are present, then all starting rni ports are mapped to RN-D nodes and then the RN-I nodes. For example, for CMN600AE with two RN-D nodes, one RN-I node, and given each RN-I or RN-D node controls three interface ports, `pvbuss_s_rni[0-2]` maps to RND0, `pvbuss_s_rni[3-5]` maps to RND1 and `pvbuss_s_rni[6-8]` maps to RNI0. Similarly, SN-F and SBSX nodes are mapped to `pvbuss_m_snf[]` ports where starting ports are mapped to SN-F and then SBSX nodes.

- Mapping HN\*\_SLC\_SIZE\_PARAM values to cache sizes:

**-8**

0KB when HN\*\_SLC\_NUM\_WAYS\_PARAM=16



Note

This value is not supported by the model.

**-2**

128KB when HN\*\_SLC\_NUM\_WAYS\_PARAM=16

**-1**

256KB when HN\*\_SLC\_NUM\_WAYS\_PARAM=16

**0**

512KB when HN\*\_SLC\_NUM\_WAYS\_PARAM=16

**1**

1MB when HN\*\_SLC\_NUM\_WAYS\_PARAM=16

**2**

2MB when HN\*\_SLC\_NUM\_WAYS\_PARAM=16

**3**

3MB when HN\*\_SLC\_NUM\_WAYS\_PARAM=12

**3**

4MB when HN\*\_SLC\_NUM\_WAYS\_PARAM=16

- Maximum number of nodes that have been verified are:
  - 7 RN-Fs
  - 2 RN-Ds
  - 3 RN-Is
  - 1 HN-I
  - 4 HN-Fs
  - 1 SN-F
- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.

- Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.
- When `HNF_ABF_PR.abf_mode` is Reserved and `HNF_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
- Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
- The optional interrupt `INTREQPPU` is not supported.
- This model implements the following AE-specific features:
  - Memory Protection Unit (MPU), with limitations listed in the Model limitations section.
  - FuSa error logging and reporting using a Fault Management Unit (FMU) and Fault Detection and Control (FDC).
  - Dedicated APB interface into FMU for fault diagnostics and control.

## Model limitations

- Out of scope:
    - PMU counters are not supported. Counter registers are implemented as **RAZ**.
    - QoS is not supported and all related registers are **RAZ/WI**.
    - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
    - Power, clock, and interrupt signals are not supported.
- The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.
- Protocol credits and flit buffers are not supported.
  - Snoop filtering is not supported.
  - Prefetch Target operations are not supported.
  - The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.
  - An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
  - Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-Lite port have not been tested.
  - RNSAMs external to the mesh network are not functionally supported.



Note

Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31])

of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address [MAX:12] instead of the actual address [MAX:6], due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address [MAX:6], but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.
- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.
- SLC/LCC partitioning dynamic allocation is not supported.
- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mvp_device_port_connect_info` register corresponding to the number of device credit slices and CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0
0x00000002000 - 0x00000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- Transactions to addresses unused by Device registers in the Configuration Register Space are always routed to the Configuration Space even when the SAM is configured to use those addresses.
- GIC communication over A4S ports is not supported.
- MPU with data coherency is not supported.
- MPU located in CXRH is not supported.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.
- For Power State Transition related flush/disable of the SLC due to HNF\_PPU\_PWPR register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.
- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.

- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- There is no support for RAS.
- The following limitations apply to System Cache Groups and Hash Target Groups:
  - A mix of local and remote HN-F targets are not supported.
  - CCG/CXRH target IDs in the HN target ID table are not supported (`sys_cache_grp_hn_nodeid_reg`).
  - The `hashed_target_grp_hnp_nodeid_reg` which is used for CCG/HN-P target IDs is not supported.
  - System/Hash Target Groups only support HN-Fs.
  - AXID hashing across HN-P/CCGs is not supported.
- For RN-D nodes, when software writes SYSCOREQ, DVM propagation gets enabled but SYSCOACK is not set.

### About the `debug_force_snoop` parameter

The CMN interconnect normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream rnf ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



This parameter is for debug purposes only. Do not use it instead of correctly configuring CMN.

## Iris and MTI instances for CMN600AE

This model has the following Iris instances:

**Table 3-898: CMN600AE Iris instances**

InstanceName	ComponentName
CMN600AE	CMN600AE
CMN600AE.bus_slave_ocm_NS	PVBusSlave
CMN600AE.bus_slave_ocm_S	PVBusSlave
CMN600AE.cmn600_cache	PVCache
CMN600AE.cmn600_cache.upstream[0]	PVBusSlave
CMN600AE.cmn600_cache.upstream[10]	PVBusSlave
CMN600AE.cmn600_cache.upstream[11]	PVBusSlave
CMN600AE.cmn600_cache.upstream[12]	PVBusSlave
CMN600AE.cmn600_cache.upstream[13]	PVBusSlave
CMN600AE.cmn600_cache.upstream[14]	PVBusSlave
CMN600AE.cmn600_cache.upstream[15]	PVBusSlave
CMN600AE.cmn600_cache.upstream[16]	PVBusSlave
CMN600AE.cmn600_cache.upstream[17]	PVBusSlave
CMN600AE.cmn600_cache.upstream[18]	PVBusSlave
CMN600AE.cmn600_cache.upstream[1]	PVBusSlave
CMN600AE.cmn600_cache.upstream[2]	PVBusSlave
CMN600AE.cmn600_cache.upstream[3]	PVBusSlave
CMN600AE.cmn600_cache.upstream[4]	PVBusSlave
CMN600AE.cmn600_cache.upstream[5]	PVBusSlave
CMN600AE.cmn600_cache.upstream[6]	PVBusSlave
CMN600AE.cmn600_cache.upstream[7]	PVBusSlave
CMN600AE.cmn600_cache.upstream[8]	PVBusSlave
CMN600AE.cmn600_cache.upstream[9]	PVBusSlave
CMN600AE.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN600AE.hnf_exclusive_monitor.bus_mapper	PVBusMapper



InstanceName	ComponentName
CMN600AE.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN600AE.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN600AE.hni_exclusive_monitor1	PVBusExclusiveMonitor
CMN600AE.hni_exclusive_monitor1.bus_mapper	PVBusMapper
CMN600AE.mpu	PVBusMapper
CMN600AE.ocm_decoder	PVBusMapper
CMN600AE.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN600AE.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN600AE.snf_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-899: CMN600AE MTI instances**

InstanceName	ComponentName
CMN600AE	CMN600AE
CMN600AE.bus_slave_ocm_NS	PVBusSlave
CMN600AE.bus_slave_ocm_S	PVBusSlave
CMN600AE.cmn600_cache	CMN600Cache
CMN600AE.cmn600_cache.upstream[0]	PVBusSlave
CMN600AE.cmn600_cache.upstream[10]	PVBusSlave
CMN600AE.cmn600_cache.upstream[11]	PVBusSlave
CMN600AE.cmn600_cache.upstream[12]	PVBusSlave
CMN600AE.cmn600_cache.upstream[13]	PVBusSlave
CMN600AE.cmn600_cache.upstream[14]	PVBusSlave
CMN600AE.cmn600_cache.upstream[15]	PVBusSlave
CMN600AE.cmn600_cache.upstream[16]	PVBusSlave
CMN600AE.cmn600_cache.upstream[17]	PVBusSlave
CMN600AE.cmn600_cache.upstream[18]	PVBusSlave
CMN600AE.cmn600_cache.upstream[1]	PVBusSlave
CMN600AE.cmn600_cache.upstream[2]	PVBusSlave
CMN600AE.cmn600_cache.upstream[3]	PVBusSlave
CMN600AE.cmn600_cache.upstream[4]	PVBusSlave
CMN600AE.cmn600_cache.upstream[5]	PVBusSlave
CMN600AE.cmn600_cache.upstream[6]	PVBusSlave
CMN600AE.cmn600_cache.upstream[7]	PVBusSlave
CMN600AE.cmn600_cache.upstream[8]	PVBusSlave
CMN600AE.cmn600_cache.upstream[9]	PVBusSlave
CMN600AE.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN600AE.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN600AE.hni_exclusive_monitor0	PVBusExclusiveMonitor

InstanceName	ComponentName
CMN600AE.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN600AE.hni_exclusive_monitor1	PVBusExclusiveMonitor
CMN600AE.hni_exclusive_monitor1.bus_mapper	PVBusMapper
CMN600AE.mpu	PVBusMapper
CMN600AE.ocm_decoder	PVBusMapper
CMN600AE.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN600AE.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN600AE.snf_mapper	PVBusMapper

## Ports for CMN600AE

**Table 3-900: Ports**

Name	Protocol	Type	Description
event_clusters	Signal	Peer	CPU event communication signal from the clusters.
event_downstream_link_signal[2]	Signal	Master	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.
event_upstream_link_signal[2]	Signal	Slave	Event from the Hub towards the CMN
fmu_eri	Signal	Master	FMU signal for critical errors
fmu_fhi	Signal	Master	FMU signal for non-critical errors
pvbus_m_cml	PVBus	Master	CML downstream ports
pvbus_m_cml_cfg	PVBus	Master	CML downstream hub configuration port
pvbus_m_hni[4]	PVBus	Master	HNI downstream ports.
pvbus_m_snf[4]	PVBus	Master	SNF downstream port.
pvbus_s_apb	PVBus	Slave	APB interface port.
pvbus_s_cml	PVBus	Slave	CML upstream ports
pvbus_s_rnf[8]	PVBus	Slave	RNF upstream ports.
pvbus_s_rni[24]	PVBus	Slave	RNI upstream ports.
reset_in	Signal	Slave	Reset signal.
rnf_sci_s[8]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in[8]	Signal	Slave	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s[24]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in[24]	Signal	Slave	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

## Parameters for CMN600AE

### **acchannelen\_rnf**

**Type**

string

**Default value**

"0"

**Note**

DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

---

For each `rnf` port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`

### **acchannelen\_rni**

**Type**

string

**Default value**

"0"

**Note**

DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

---

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`

### **cache\_state\_modelled**

**Type**

bool

**Default value**

0x0

**Description**

Model the cache state.

**debug\_force\_snoop****Type**

bool

**Default value**

false

The CMN600AE interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.



This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

**disable\_CML\_port****Type**

bool

**Default value**

0x0

**Description**

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enable PVBUSLoggers for the downstream ports in the CMN model.

**enable\_rnsam\_to\_hnf\_wider\_hash****Type**

bool

**Default value**

false

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

**fdc\_key****Type**

uint8\_t

**Default value**

0

`por_fdc_key` register value is checked against this key.

**fm\_u\_key****Type**

uint8\_t

**Default value**

0

`por_fm_u_key` register value is checked against this key.

**force\_rnsam\_internal****Type**

bool

**Default value**

0x1

**Description**

Force all RNSAMs to be internal independently of the mesh topology.

**mesh\_config\_file****Type**

string

**Default value**

""

**Description**

Name of a file containing mesh placement of CMN600AE components.

**number\_of\_mpu\_programmable\_regions****Type**

uint32\_t

**Default value**

32

Number of MPU programmable regions.

Valid values are 0, 8, 16 and 32.

**periphbase****Type**

uint64\_t

**Default value**

0x20000000

Value for PERIPHBASE. Bits [25:0] are treated 0

**print\_cmn\_ccix\_config****Type**

bool

**Default value**

0x0

**Description**

Print information about the CCIX configuration.

**print\_cmn\_config****Type**

bool

**Default value**

0x0

**Description**

Print the mesh topology and children pointers acquired from the YAML file.

**revision****Type**

string

**Default value**

"r1p0"

Component revision.

Currently supports r1p0.

**rnf\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

**rni\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

**show\_banner****Type**

uint64\_t

**Default value**

2

Show component banner:

- 0
- supress entire banner
- 1
- suppress config file
- 2+
- show full banner

**skip\_cmn\_config\_check**

**Type**

bool

**Default value**

0x0

**Description**

Skip any topology configuration checks. The maximum number of devices per type not verified.

**use\_yaml\_periphbase**

**Type**

bool

**Default value**

false

Use yaml param CFGM\_PERIPHBASE\_PARAM to specify periphbase address.

If false, model parameter periphbase will be used.

3.10.16 CMN600CMLHub

CMN600 CML Interconnect Hub Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-901: IP revisions support

Revision	Quality level
N/A	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Iris and MTI instances for CMN600CMLHub**

This model has the following Iris instances:



**Table 3-902: CMN600CMLHub Iris instances**

InstanceName	ComponentName
CMN600CMLHub	CMN600CMLHub
CMN600CMLHub.CMN600CMLHubCache	PVCache
CMN600CMLHub.CMN600CMLHubCache.upstream[0]	PVBusSlave
CMN600CMLHub.CMN600CMLHubCache.upstream[1]	PVBusSlave
CMN600CMLHub.CMN600CMLHubCache.upstream[2]	PVBusSlave
CMN600CMLHub.CMN600CMLHubCache.upstream[3]	PVBusSlave
CMN600CMLHub.bus_s_cfg	PVBusSlave
CMN600CMLHub.cache_downstream_exclusive_monitor0	PVBusExclusiveMonitor
CMN600CMLHub.cache_downstream_exclusive_monitor0.bus_mapper	PVBusMapper
CMN600CMLHub.cache_downstream_exclusive_monitor1	PVBusExclusiveMonitor
CMN600CMLHub.cache_downstream_exclusive_monitor1.bus_mapper	PVBusMapper
CMN600CMLHub.cache_downstream_exclusive_monitor2	PVBusExclusiveMonitor
CMN600CMLHub.cache_downstream_exclusive_monitor2.bus_mapper	PVBusMapper
CMN600CMLHub.cache_downstream_exclusive_monitor3	PVBusExclusiveMonitor
CMN600CMLHub.cache_downstream_exclusive_monitor3.bus_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-903: CMN600CMLHub MTI instances**

InstanceName	ComponentName
CMN600CMLHub	CMN600CML
CMN600CMLHub.CMN600CMLHubCache	PVCache
CMN600CMLHub.CMN600CMLHubCache.upstream[0]	PVBusSlave
CMN600CMLHub.CMN600CMLHubCache.upstream[1]	PVBusSlave
CMN600CMLHub.CMN600CMLHubCache.upstream[2]	PVBusSlave
CMN600CMLHub.CMN600CMLHubCache.upstream[3]	PVBusSlave
CMN600CMLHub.bus_s_cfg	PVBusSlave
CMN600CMLHub.cache_downstream_exclusive_monitor0	PVBusExclusiveMonitor
CMN600CMLHub.cache_downstream_exclusive_monitor0.bus_mapper	PVBusMapper
CMN600CMLHub.cache_downstream_exclusive_monitor1	PVBusExclusiveMonitor
CMN600CMLHub.cache_downstream_exclusive_monitor1.bus_mapper	PVBusMapper
CMN600CMLHub.cache_downstream_exclusive_monitor2	PVBusExclusiveMonitor
CMN600CMLHub.cache_downstream_exclusive_monitor2.bus_mapper	PVBusMapper
CMN600CMLHub.cache_downstream_exclusive_monitor3	PVBusExclusiveMonitor
CMN600CMLHub.cache_downstream_exclusive_monitor3.bus_mapper	PVBusMapper

## Ports for CMN600CMLHub

**Table 3-904: Ports**

Name	Protocol	Type	Description
event_downstream_link_signal[4]	Signal	Slave	CPU downstream event communication signal.
event_upstream_link_signal[4]	Signal	Master	CPU upstream event communication signal.
pvbust_m[4]	PVBus	Master	Downstream CCIX port.
pvbust_s[4]	PVBus	Slave	Upstream CCIX ports.
pvbust_s_cfg[4]	PVBus	Slave	Upstream config ports.
reset_signal	Signal	Slave	Reset signal.

## Parameters for CMN600CMLHub

### cache\_state\_modelled

#### Type

bool

#### Default value

0x0

#### Description

Model the cache state.

### enable\_logger

#### Type

bool

#### Default value

0x0

#### Description

Enable PVBusLoggers for the downstream ports in the CMN model.

## 3.10.17 CMN650

CMN650 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-905: IP revisions support**

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About CMN650

- Major IP revisions (rX) are modeled and are controlled by the `revision` parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of the CHI nodes. Set it to the name of the yml configuration file emitted by Socrates. You must use version r1p4-01rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.

The model supports the following features:

- `rnf`, `rni/rnd`, `hni`, and `snf/sbsx` interface ports. The mapping between the port number and `NodeId` is based on the `NodeId` index. For example, RNF2 controls `pvtbus_s_rnf[2]`. Its index is specified in the `node_info` register as `logical_id`. Similar behavior can be expected for HN-I.

If both RN-D and RN-I nodes are present, then all starting `rni` ports are mapped to RN-D nodes and then the RN-I nodes. For example, for CMN650 with two RN-D nodes, one RN-I node, and given each RN-I or RN-D node controls three interface ports, `pvtbus_s_rni[0-2]` maps to RND0, `pvtbus_s_rni[3-5]` maps to RND1 and `pvtbus_s_rni[6-8]` maps to RNI0.

Similarly, SN-F and SBSX nodes are mapped to `pvtbus_m_snf[]` ports where starting ports are mapped to SN-F and then SBSX nodes.

- Mapping HN\*\_SLC\_SIZE\_PARAM values to cache sizes:

-8

OKB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16



Note

This value is not supported by the model.

-2

128KB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16

-1

256KB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16

0

512KB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16

1

1MB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16

2

2MB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16

**3**

3MB where `HN*_SLC_NUM_WAYS_PARAM=12`

**3**

4MB where `HN*_SLC_NUM_WAYS_PARAM=16`

- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
  - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.
  - When `HN*_ABF_PR.abf_mode` is Reserved and `HN*_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
  - Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
  - The optional interrupt `INTREQPPU` is not supported.

## Model limitations

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
  - Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.
- Non-XY routing behavior is not supported.
- The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.
- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-lite port have not been tested.
- RNSAMs external to the mesh network are not functionally supported.



Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31]) of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address [MAX:12] instead of the actual address [MAX:6], due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address [MAX:6], but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.
- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.
- SLC/LCC partitioning dynamic allocation is not supported.
- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mvp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0
0x00000002000 - 0x00000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- Transactions to addresses unused by Device registers in the Configuration Register Space are always routed to the Configuration Space even when the SAM is configured to use those addresses.
- GIC communication over A4S ports is not supported.
- MPAM features are not supported in the model, and the configuration-dependent register reset values do not match the RTL. Some of the registers (`por_hnf_mpam_idr` to `por_hnf_mpam_mbwumon_idr`) shared between security modes are not present in MPAM\_S.
- The following limitations apply to System Cache Groups and Hash Target Groups:
  - A mix of local and remote HN-F targets is not supported.
  - CCG/CXRH target IDs in the HN target ID table are not supported (`sys_cache_grp_hn_nodeid_reg`).

- The `hashed_target_grp_hnp_nodeid_reg` which is used for CCG/HN-P target IDs is not supported.
- System/Hash Target Groups only support HN-Fs.
- AXID hashing across HN-P/CCGs is not supported.
- The MPAM reset values described in `por_hnf_mpam_ns_por_hnf_mpam_idr[31:24]` default to the not-supported state. The model parameter `hnf_mpam_idr_override` can be used to override this but does not implement any of the functionality.
- MPAM\_S `secure_register_groups_override` is not supported.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.
- For Power State Transition related flush/disable of the SLC due to HNF\_PPU\_PWPR register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.
- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` are used as the default target only when `RNSAM_STATUS.use_default_node=1`. Otherwise, when a `txn` is sent into the model that does not belong to any of the address regions programmed in the RNSAM, it is routed to the HN-D irrespective of what is programmed in the `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` fields.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- HN-F SAM limitations:
  - Address masking in default hash regions in HN-F SAM is not supported.
  - Hashing in default hash regions in HN-F SAM is not supported. However, trace `SNF_HASHING_Target_SNF` displays the target SNF as if 3SN/6SN hashing were enabled.



The transaction is always routed to the SNO nodeid programmed in the SAM\_CONTROL register.

- 
- Re-programming regions in HN-F SAM is not tested.
  - Hashing across CCGs in HN-F SAM is not supported.

- For an HTG/SCG, the model does not consider the secondary region if the primary region is not valid.
- There is no support for RAS.
- For RN-D nodes, when software writes SYSCOREQ, DVM propagation gets enabled but SYSCOACK is not set.

### About the `debug_force_snoop` parameter

The CMN interconnect normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream rnf ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



Note

This parameter is for debug purposes only. Do not use it instead of correctly configuring CMN.

---

### Iris and MTI instances for CMN650

This model has the following Iris instances:

**Table 3-906: CMN650 Iris instances**

InstanceName	ComponentName
CMN650	CMN650
CMN650.bus_slave_ocm_NS	PVBusSlave
CMN650.bus_slave_ocm_S	PVBusSlave
CMN650.cmn600_cache	PVCache
CMN650.cmn600_cache.upstream[0]	PVBusSlave
CMN650.cmn600_cache.upstream[10]	PVBusSlave
CMN650.cmn600_cache.upstream[11]	PVBusSlave
CMN650.cmn600_cache.upstream[12]	PVBusSlave
CMN650.cmn600_cache.upstream[13]	PVBusSlave
CMN650.cmn600_cache.upstream[14]	PVBusSlave
CMN650.cmn600_cache.upstream[15]	PVBusSlave
CMN650.cmn600_cache.upstream[16]	PVBusSlave
CMN650.cmn600_cache.upstream[17]	PVBusSlave
CMN650.cmn600_cache.upstream[18]	PVBusSlave
CMN650.cmn600_cache.upstream[19]	PVBusSlave
CMN650.cmn600_cache.upstream[1]	PVBusSlave
CMN650.cmn600_cache.upstream[20]	PVBusSlave
CMN650.cmn600_cache.upstream[21]	PVBusSlave
CMN650.cmn600_cache.upstream[22]	PVBusSlave
CMN650.cmn600_cache.upstream[23]	PVBusSlave
CMN650.cmn600_cache.upstream[24]	PVBusSlave
CMN650.cmn600_cache.upstream[25]	PVBusSlave
CMN650.cmn600_cache.upstream[26]	PVBusSlave
CMN650.cmn600_cache.upstream[27]	PVBusSlave
CMN650.cmn600_cache.upstream[28]	PVBusSlave
CMN650.cmn600_cache.upstream[29]	PVBusSlave
CMN650.cmn600_cache.upstream[2]	PVBusSlave
CMN650.cmn600_cache.upstream[30]	PVBusSlave
CMN650.cmn600_cache.upstream[31]	PVBusSlave
CMN650.cmn600_cache.upstream[32]	PVBusSlave
CMN650.cmn600_cache.upstream[33]	PVBusSlave
CMN650.cmn600_cache.upstream[3]	PVBusSlave
CMN650.cmn600_cache.upstream[4]	PVBusSlave
CMN650.cmn600_cache.upstream[5]	PVBusSlave
CMN650.cmn600_cache.upstream[6]	PVBusSlave
CMN650.cmn600_cache.upstream[7]	PVBusSlave
CMN650.cmn600_cache.upstream[8]	PVBusSlave
CMN650.cmn600_cache.upstream[9]	PVBusSlave



InstanceName	ComponentName
CMN650.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN650.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN650.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN650.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN650.hni_exclusive_monitor1	PVBusExclusiveMonitor
CMN650.hni_exclusive_monitor1.bus_mapper	PVBusMapper
CMN650.hni_exclusive_monitor2	PVBusExclusiveMonitor
CMN650.hni_exclusive_monitor2.bus_mapper	PVBusMapper
CMN650.ocm_decoder	PVBusMapper
CMN650.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN650.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN650.snf_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-907: CMN650 MTI instances**

InstanceName	ComponentName
CMN650	CMN650
CMN650.bus_slave_ocm_NS	PVBusSlave
CMN650.bus_slave_ocm_S	PVBusSlave
CMN650.cmn600_cache	CMN600Cache
CMN650.cmn600_cache.upstream[0]	PVBusSlave
CMN650.cmn600_cache.upstream[10]	PVBusSlave
CMN650.cmn600_cache.upstream[11]	PVBusSlave
CMN650.cmn600_cache.upstream[12]	PVBusSlave
CMN650.cmn600_cache.upstream[13]	PVBusSlave
CMN650.cmn600_cache.upstream[14]	PVBusSlave
CMN650.cmn600_cache.upstream[15]	PVBusSlave
CMN650.cmn600_cache.upstream[16]	PVBusSlave
CMN650.cmn600_cache.upstream[17]	PVBusSlave
CMN650.cmn600_cache.upstream[18]	PVBusSlave
CMN650.cmn600_cache.upstream[19]	PVBusSlave
CMN650.cmn600_cache.upstream[1]	PVBusSlave
CMN650.cmn600_cache.upstream[20]	PVBusSlave
CMN650.cmn600_cache.upstream[21]	PVBusSlave
CMN650.cmn600_cache.upstream[22]	PVBusSlave
CMN650.cmn600_cache.upstream[23]	PVBusSlave
CMN650.cmn600_cache.upstream[24]	PVBusSlave
CMN650.cmn600_cache.upstream[25]	PVBusSlave
CMN650.cmn600_cache.upstream[26]	PVBusSlave

InstanceName	ComponentName
CMN650.cmn600_cache.upstream[27]	PVBusSlave
CMN650.cmn600_cache.upstream[28]	PVBusSlave
CMN650.cmn600_cache.upstream[29]	PVBusSlave
CMN650.cmn600_cache.upstream[2]	PVBusSlave
CMN650.cmn600_cache.upstream[30]	PVBusSlave
CMN650.cmn600_cache.upstream[31]	PVBusSlave
CMN650.cmn600_cache.upstream[32]	PVBusSlave
CMN650.cmn600_cache.upstream[33]	PVBusSlave
CMN650.cmn600_cache.upstream[3]	PVBusSlave
CMN650.cmn600_cache.upstream[4]	PVBusSlave
CMN650.cmn600_cache.upstream[5]	PVBusSlave
CMN650.cmn600_cache.upstream[6]	PVBusSlave
CMN650.cmn600_cache.upstream[7]	PVBusSlave
CMN650.cmn600_cache.upstream[8]	PVBusSlave
CMN650.cmn600_cache.upstream[9]	PVBusSlave
CMN650.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN650.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN650.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN650.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN650.hni_exclusive_monitor1	PVBusExclusiveMonitor
CMN650.hni_exclusive_monitor1.bus_mapper	PVBusMapper
CMN650.hni_exclusive_monitor2	PVBusExclusiveMonitor
CMN650.hni_exclusive_monitor2.bus_mapper	PVBusMapper
CMN650.ocm_decoder	PVBusMapper
CMN650.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN650.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN650.snf_mapper	PVBusMapper

## Ports for CMN650

**Table 3-908: Ports**

Name	Protocol	Type	Description
event_clusters	Signal	Peer	CPU event communication signal from the clusters.
event_downstream_link_signal[10]	Signal	Master	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.
event_upstream_link_signal[10]	Signal	Slave	Event from the Hub towards the CMN
pvbus_m_cml	PVBus	Master	CML downstream ports
pvbus_m_cml_cfg	PVBus	Master	CML downstream hub configuration port
pvbus_m_hni[16]	PVBus	Master	HNI downstream ports.

Name	Protocol	Type	Description
pvbus_m_snf[40]	PVBus	Master	SNF downstream port.
pvbus_s_apb	PVBus	Slave	APB interface port.
pvbus_s_cml	PVBus	Slave	CML upstream ports
pvbus_s_rnf[64]	PVBus	Slave	RNF upstream ports.
pvbus_s_rni[96]	PVBus	Slave	RNI upstream ports.
reset_in	Signal	Slave	Reset signal.
rnf_sci_s[64]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in[64]	Signal	Slave	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s[96]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in[96]	Signal	Slave	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

## Parameters for CMN650

### acchannelen\_rnf

#### Type

string

#### Default value

"0"



DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each rnf port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`

### acchannelen\_rni

#### Type

string

#### Default value

"0"

**Note**

DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

---

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`

### **cache\_state\_modelled**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Model the cache state.

### **debug\_force\_snoop**

#### **Type**

bool

#### **Default value**

false

The CMN650 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

**Note**

This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

### **disable\_CML\_port**

#### **Type**

bool

**Default value**

0x0

**Description**

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enable PVBUSLoggers for the downstream ports in the CMN model.

**enable\_rnsam\_to\_hnf\_wider\_hash****Type**

bool

**Default value**

false

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

**force\_rnsam\_internal****Type**

bool

**Default value**

0x1

**Description**

Force all RNSAMs to be internal independently of the mesh topology.

**hnf\_mpam\_idr\_override****Type**

uint64\_t

**Default value**

0

Set to override `hnf_mpam_idr[31:24]` value. Bit[28] is Reserved and is ignored.

**mesh\_config\_file****Type**

string

**Default value**

""

**Description**

Name of a file containing mesh placement of CMN650 components.

**periphbase****Type**

uint64\_t

**Default value**

0x20000000

Value for PERIPHBASE. Bits [27:0] are treated 0

**print\_cmn\_ccix\_config****Type**

bool

**Default value**

0x0

**Description**

Print information about the CCIX configuration.

**print\_cmn\_config****Type**

bool

**Default value**

0x0

**Description**

Print the mesh topology and children pointers acquired from the YAML file.

**revision****Type**

string

**Default value**

"r1p1"

Component revision.

Currently supports r1p1.

**rnf\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

**rni\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

**show\_banner****Type**

uint64\_t

**Default value**

2

Show component banner:

**0**

supress entire banner

**1**

suppress config file

2+  
show full banner

**skip\_cmn\_config\_check**

Type  
bool

Default value  
0x0

Description  
Skip any topology configuration checks. The maximum number of devices per type not verified.

**use\_yaml\_periphbase**

Type  
bool

Default value  
false

Use yaml param CFGM\_PERIPHBASE\_PARAM to specify periphbase address.

If false, model parameter periphbase will be used.

3.10.18 CMN650R2

CMN650R2 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-909: IP revisions support

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About CMN650R2

- Major IP revisions (rX) are modeled and are controlled by the revision parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the periph\_id2 register.
- To configure the model, you must have installed Arm Socrates. The mesh\_config\_file parameter defines the mesh placement of the CHI nodes. Set it to the name of the yaml configuration file emitted by Socrates. You must use version r1p4-01rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.



- Interconnect models are based on the TRM description and do not typically model RTL defects.

The model supports the following features:

- `rnf`, `rni/rnd`, `hni`, and `snf/sbsx` interface ports. The mapping between the port number and `nodeId` is based on the `nodeId` index. For example, `RNF2` controls `pvbuss_s_rnf[2]`. Its index is specified in the `node_info` register as `logical_id`. Similar behavior can be expected for `HN-I`.

If both `RN-D` and `RN-I` nodes are present, then all starting `rni` ports are mapped to `RN-D` nodes and then the `RN-I` nodes. For example, for `CMN650` with two `RN-D` nodes, one `RN-I` node, and given each `RN-I` or `RN-D` node controls three interface ports, `pvbuss_s_rni[0-2]` maps to `RND0`, `pvbuss_s_rni[3-5]` maps to `RND1` and `pvbuss_s_rni[6-8]` maps to `RNI0`.

Similarly, `SN-F` and `SBSX` nodes are mapped to `pvbuss_m_snf[]` ports where starting ports are mapped to `SN-F` and then `SBSX` nodes.

- Mapping `HN*_SLC_SIZE_PARAM` values to cache sizes:

-8

OKB where `HN*_SLC_NUM_WAYS_PARAM=16`



Note

This value is not supported by the model.

-2

128KB where `HN*_SLC_NUM_WAYS_PARAM=16`

-1

256KB where `HN*_SLC_NUM_WAYS_PARAM=16`

0

512KB where `HN*_SLC_NUM_WAYS_PARAM=16`

1

1MB where `HN*_SLC_NUM_WAYS_PARAM=16`

2

2MB where `HN*_SLC_NUM_WAYS_PARAM=16`

3

3MB where `HN*_SLC_NUM_WAYS_PARAM=12`

3

4MB where `HN*_SLC_NUM_WAYS_PARAM=16`

- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any `HN-F` in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the `HN-F` belongs to.
  - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.

- When `HN_F_AB_F_PR.abf_mode` is Reserved and `HN_F_AB_F_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
- Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
- The optional interrupt `INTREQ_PPU` is not supported.

## Model limitations

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
  - Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.
- Non-XY routing behavior is not supported.
- The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.
- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-lite port have not been tested.
- RNSAMs external to the mesh network are not functionally supported.



Note

Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31]) of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address [MAX:12] instead of the actual address [MAX:6], due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address [MAX:6], but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.

- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.
- SLC/LCC partitioning dynamic allocation is not supported.
- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mvp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0
0x00000002000 - 0x00000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- Transactions to addresses unused by Device registers in the Configuration Register Space are always routed to the Configuration Space even when the SAM is configured to use those addresses.
- GIC communication over A4S ports is not supported.
- MPAM features are not supported in the model, and the configuration-dependent register reset values do not match the RTL. Some of the registers (`por_hnf_mpam_idr` to `por_hnf_mpam_mbwumon_idr`) shared between security modes are not present in MPAM\_S.
- The following limitations apply to System Cache Groups and Hash Target Groups:
  - A mix of local and remote HN-F targets is not supported.
  - CCG/CXRH target IDs in the HN target ID table are not supported (`sys_cache_grp_hn_nodeid_reg`).
  - The `hashed_target_grp_hnp_nodeid_reg` which is used for CCG/HN-P target IDs is not supported.
  - System/Hash Target Groups only support HN-Fs.
  - AXID hashing across HN-P/CCGs is not supported.
- The MPAM reset values described in `por_hnf_mpam_ns_por_hnf_mpam_idr[31:24]` default to the not-supported state. The model parameter `hnf_mpam_idr_override` can be used to override this but does not implement any of the functionality.
- MPAM\_S `secure_register_groups_override` is not supported.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.

- For Power State Transition related flush/disable of the SLC due to HNF\_PPU\_PWPR register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.
- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` are used as the default target only when `RNSAM_STATUS.use_default_node=1`. Otherwise, when a txn is sent into the model that does not belong to any of the address regions programmed in the RNSAM, it is routed to the HN-D irrespective of what is programmed in the `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` fields.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- HN-F SAM limitations:
  - Address masking in default hash regions in HN-F SAM is not supported.
  - Hashing in default hash regions in HN-F SAM is not supported. However, trace `SNF_HASHING_Target_SNF` displays the target SNF as if 3SN/6SN hashing were enabled.



Note

The transaction is always routed to the SNO nodeid programmed in the `SAM_CONTROL` register.

- Re-programming regions in HN-F SAM is not tested.
- Hashing across CCGs in HN-F SAM is not supported.
- CAL2 support for HN-P and RN-D is not tested.
- The maximum number of 256 RN-Fs is not verified. 74 is the largest number tested.
- The maximum number of 40 SNs is not verified. 20 is the largest number tested.
- The maximum number of 36 RN-Is is not verified. 16 is the largest number tested.
- The maximum number of 16 HN-Is is not verified. 5 is the largest number tested.
- Early DVM completion is not supported.
- CCIX port to port forwarding is not supported.
- No support for up to 512 CXRAs with no RAID aliasing and 256 RN-Fs on a single chip.

- Each HN-F can support tracking of up to 512 logical processors for exclusive operations. However, the value of the RO field `num_excl` in the HN-F unit info register cannot exceed 255.
- For RN-D nodes, when software writes `SYSCOREQ`, DVM propagation gets enabled but `SYSCOACK` is not set.

### About the `debug_force_snoop` parameter

The CMN interconnect normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream rnf ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



This parameter is for debug purposes only. Do not use it instead of correctly configuring CMN.

---

### Iris and MTI instances for CMN650R2

This model has the following Iris instances:

**Table 3-910: CMN650R2 Iris instances**

InstanceName	ComponentName
CMN650R2	CMN650R2
CMN650R2.bus_slave_ocm_NS	PVBusSlave
CMN650R2.bus_slave_ocm_S	PVBusSlave
CMN650R2.cmn600_cache	PVCache
CMN650R2.cmn600_cache.upstream[0]	PVBusSlave
CMN650R2.cmn600_cache.upstream[10]	PVBusSlave
CMN650R2.cmn600_cache.upstream[11]	PVBusSlave
CMN650R2.cmn600_cache.upstream[12]	PVBusSlave
CMN650R2.cmn600_cache.upstream[13]	PVBusSlave
CMN650R2.cmn600_cache.upstream[14]	PVBusSlave
CMN650R2.cmn600_cache.upstream[15]	PVBusSlave
CMN650R2.cmn600_cache.upstream[16]	PVBusSlave
CMN650R2.cmn600_cache.upstream[17]	PVBusSlave
CMN650R2.cmn600_cache.upstream[18]	PVBusSlave
CMN650R2.cmn600_cache.upstream[19]	PVBusSlave
CMN650R2.cmn600_cache.upstream[1]	PVBusSlave
CMN650R2.cmn600_cache.upstream[20]	PVBusSlave
CMN650R2.cmn600_cache.upstream[21]	PVBusSlave
CMN650R2.cmn600_cache.upstream[22]	PVBusSlave
CMN650R2.cmn600_cache.upstream[23]	PVBusSlave
CMN650R2.cmn600_cache.upstream[24]	PVBusSlave
CMN650R2.cmn600_cache.upstream[25]	PVBusSlave
CMN650R2.cmn600_cache.upstream[26]	PVBusSlave
CMN650R2.cmn600_cache.upstream[27]	PVBusSlave
CMN650R2.cmn600_cache.upstream[28]	PVBusSlave
CMN650R2.cmn600_cache.upstream[29]	PVBusSlave
CMN650R2.cmn600_cache.upstream[2]	PVBusSlave
CMN650R2.cmn600_cache.upstream[30]	PVBusSlave
CMN650R2.cmn600_cache.upstream[31]	PVBusSlave
CMN650R2.cmn600_cache.upstream[32]	PVBusSlave
CMN650R2.cmn600_cache.upstream[33]	PVBusSlave
CMN650R2.cmn600_cache.upstream[3]	PVBusSlave
CMN650R2.cmn600_cache.upstream[4]	PVBusSlave
CMN650R2.cmn600_cache.upstream[5]	PVBusSlave
CMN650R2.cmn600_cache.upstream[6]	PVBusSlave
CMN650R2.cmn600_cache.upstream[7]	PVBusSlave
CMN650R2.cmn600_cache.upstream[8]	PVBusSlave
CMN650R2.cmn600_cache.upstream[9]	PVBusSlave

InstanceName	ComponentName
CMN650R2.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN650R2.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN650R2.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN650R2.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN650R2.hni_exclusive_monitor1	PVBusExclusiveMonitor
CMN650R2.hni_exclusive_monitor1.bus_mapper	PVBusMapper
CMN650R2.hni_exclusive_monitor2	PVBusExclusiveMonitor
CMN650R2.hni_exclusive_monitor2.bus_mapper	PVBusMapper
CMN650R2.ocm_decoder	PVBusMapper
CMN650R2.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN650R2.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN650R2.snf_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-911: CMN650R2 MTI instances**

InstanceName	ComponentName
CMN650R2	CMN650R2
CMN650R2.bus_slave_ocm_NS	PVBusSlave
CMN650R2.bus_slave_ocm_S	PVBusSlave
CMN650R2.cmn600_cache	CMN600Cache
CMN650R2.cmn600_cache.upstream[0]	PVBusSlave
CMN650R2.cmn600_cache.upstream[10]	PVBusSlave
CMN650R2.cmn600_cache.upstream[11]	PVBusSlave
CMN650R2.cmn600_cache.upstream[12]	PVBusSlave
CMN650R2.cmn600_cache.upstream[13]	PVBusSlave
CMN650R2.cmn600_cache.upstream[14]	PVBusSlave
CMN650R2.cmn600_cache.upstream[15]	PVBusSlave
CMN650R2.cmn600_cache.upstream[16]	PVBusSlave
CMN650R2.cmn600_cache.upstream[17]	PVBusSlave
CMN650R2.cmn600_cache.upstream[18]	PVBusSlave
CMN650R2.cmn600_cache.upstream[19]	PVBusSlave
CMN650R2.cmn600_cache.upstream[1]	PVBusSlave
CMN650R2.cmn600_cache.upstream[20]	PVBusSlave
CMN650R2.cmn600_cache.upstream[21]	PVBusSlave
CMN650R2.cmn600_cache.upstream[22]	PVBusSlave
CMN650R2.cmn600_cache.upstream[23]	PVBusSlave
CMN650R2.cmn600_cache.upstream[24]	PVBusSlave
CMN650R2.cmn600_cache.upstream[25]	PVBusSlave
CMN650R2.cmn600_cache.upstream[26]	PVBusSlave

InstanceName	ComponentName
CMN650R2.cmn600_cache.upstream[27]	PVBusSlave
CMN650R2.cmn600_cache.upstream[28]	PVBusSlave
CMN650R2.cmn600_cache.upstream[29]	PVBusSlave
CMN650R2.cmn600_cache.upstream[2]	PVBusSlave
CMN650R2.cmn600_cache.upstream[30]	PVBusSlave
CMN650R2.cmn600_cache.upstream[31]	PVBusSlave
CMN650R2.cmn600_cache.upstream[32]	PVBusSlave
CMN650R2.cmn600_cache.upstream[33]	PVBusSlave
CMN650R2.cmn600_cache.upstream[3]	PVBusSlave
CMN650R2.cmn600_cache.upstream[4]	PVBusSlave
CMN650R2.cmn600_cache.upstream[5]	PVBusSlave
CMN650R2.cmn600_cache.upstream[6]	PVBusSlave
CMN650R2.cmn600_cache.upstream[7]	PVBusSlave
CMN650R2.cmn600_cache.upstream[8]	PVBusSlave
CMN650R2.cmn600_cache.upstream[9]	PVBusSlave
CMN650R2.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN650R2.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN650R2.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN650R2.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN650R2.hni_exclusive_monitor1	PVBusExclusiveMonitor
CMN650R2.hni_exclusive_monitor1.bus_mapper	PVBusMapper
CMN650R2.hni_exclusive_monitor2	PVBusExclusiveMonitor
CMN650R2.hni_exclusive_monitor2.bus_mapper	PVBusMapper
CMN650R2.ocm_decoder	PVBusMapper
CMN650R2.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN650R2.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN650R2.snf_mapper	PVBusMapper

## Ports for CMN650R2

**Table 3-912: Ports**

Name	Protocol	Type	Description
event_clusters	Signal	Peer	CPU event communication signal from the clusters.
event_downstream_link_signal[10]	Signal	Master	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.
event_upstream_link_signal[10]	Signal	Slave	Event from the Hub towards the CMN
pvbus_m_cml	PVBus	Master	CML downstream ports
pvbus_m_cml_cfg	PVBus	Master	CML downstream hub configuration port
pvbus_m_hni[16]	PVBus	Master	HNI downstream ports.



Name	Protocol	Type	Description
pvbus_m_snf[40]	PVBus	Master	SNF downstream port.
pvbus_s_apb	PVBus	Slave	APB interface port.
pvbus_s_cml	PVBus	Slave	CML upstream ports
pvbus_s_rnf[256]	PVBus	Slave	RNF upstream ports.
pvbus_s_rni[108]	PVBus	Slave	RNI upstream ports.
reset_in	Signal	Slave	Reset signal.
rnf_sci_s[256]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in[256]	Signal	Slave	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s[108]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in[108]	Signal	Slave	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

## Parameters for CMN650R2

### **acchannelen\_rnf**

#### Type

string

#### Default value

"0"



DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each rnf port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`

### **acchannelen\_rni**

#### Type

string

#### Default value

"0"

**Note**

DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

---

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`

### **cache\_state\_modelled**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Model the cache state.

### **debug\_force\_snoop**

#### **Type**

bool

#### **Default value**

false

The CMN650R2 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

**Note**

This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

### **disable\_CML\_port**

#### **Type**

bool

**Default value**

0x0

**Description**

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enable PVBUSLoggers for the downstream ports in the CMN model.

**enable\_rnsam\_to\_hnf\_wider\_hash****Type**

bool

**Default value**

false

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

**force\_rnsam\_internal****Type**

bool

**Default value**

0x1

**Description**

Force all RNSAMs to be internal independently of the mesh topology.

**hnf\_mpam\_idr\_override****Type**

uint64\_t

**Default value**

0

Set to override `hnf_mpam_idr[31:24]` value. Bit[28] is Reserved and is ignored.

**mesh\_config\_file****Type**

string

**Default value**

""

**Description**

Name of a file containing mesh placement of CMN650R2 components.

**periphbase****Type**

uint64\_t

**Default value**

0x20000000

Value for PERIPHBASE. Bits [27:0] are treated 0

**print\_cmn\_ccix\_config****Type**

bool

**Default value**

0x0

**Description**

Print information about the CCIX configuration.

**print\_cmn\_config****Type**

bool

**Default value**

0x0

**Description**

Print the mesh topology and children pointers acquired from the YAML file.

**revision****Type**

string

**Default value**

"r2p0"

Component revision.

Currently supports r2p0.

**rnf\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

**rni\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

**show\_banner****Type**

uint64\_t

**Default value**

2

Show component banner:

**0**

supress entire banner

**1**

suppress config file

2+  
show full banner

**skip\_cmn\_config\_check**

Type  
bool

Default value  
0x0

Description  
Skip any topology configuration checks. The maximum number of devices per type not verified.

**use\_yaml\_periphbase**

Type  
bool

Default value  
false

Use yaml param CFGM\_PERIPHBASE\_PARAM to specify periphbase address.

If false, model parameter periphbase will be used.

3.10.19 CMN700

CMN700 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-913: IP revisions support

Revision	Quality level
r0p0	Full support
r1p0	Full support
r2p0	Full support
r3p0	Full support
r3p3	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Changes in 11.29.19

Ports added:

- pvbus\_m\_dmc\_axu
- pvbus\_m\_dsu\_axu

Parameters added:

- `dmc_periphbase`
- `dsu_periphbase`

## About CMN700

- Major IP revisions (rX) are modeled and are controlled by the `revision` parameter or the topology file.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of the CHI nodes. Set it to the name of the yml configuration file emitted by Socrates. You must use version SYSOC-BN-00001 r1p6-02lac1 of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.
- The maximum mesh size supported is X=12, Y=12.
- The mapping between the port number for `rnf`, `rni/rnd`, `hni`, and `snf/sbsx` interface ports and `NodeId` is based on the `NodeId` index. For example, RNF2 controls `pvbuss_s_rnf[2]`. Its index is specified in the `node_info` register as `logical_id`. Similar behavior can be expected for HN-I.

If both RN-D and RN-I nodes are present, then all starting `rni` ports are mapped to RN-D nodes and then the RN-I nodes. For example, for CMN700 with two RN-D nodes, one RN-I node, and given each RN-I or RN-D node controls three interface ports, `pvbuss_s_rni[0-2]` maps to `RND0`, `pvbuss_s_rni[3-5]` maps to `RND1` and `pvbuss_s_rni[6-8]` maps to `RNI0`.

Similarly, SN-F and SBSX nodes are mapped to `pvbuss_m_snf[]` ports where starting ports are mapped to SN-F and then SBSX nodes.

- See the CMN700r3 TRM for the `HN*_SLC_SIZE_PARAM` values.
- There is limited support for RNSAMs external to the CMN. See the Model limitations section for more information.
- CCG device id is CCG id + 1 when `CAL2` and `PCIE_ENABLE` are set for port 1.
- CXL Type-3 (CXL.mem) devices can be connected to the `pvbuss_m_cxs[]` ports. The CXRH nodes, if any, are connected to `pvbuss_m_cxs[0]` ... followed by the CCG nodes, if any.
- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
  - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.
  - When `HN*_ABF_PR.abf_mode` is Reserved and `HN*_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.

- Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
- The optional interrupt `INTREQFPU` is not supported.
- There is limited support for RAS:
  - Error logging and reporting functionality for HN-I, SBSX, XP, and MTU are supported.
  - RAS-related interrupts (`INTREQERRS`, `INTREQFAULTS`, `INTREQFAULTNS`, `INTREQERRNS`) have been added.
  - Central RAS interrupt-handling functionality of HN-D is supported.



Enabling RAS can impact performance. To avoid this, RAS is disabled by default in the CMN700 model. To enable RAS, set the `enable_ras` parameter to true.

- There is support for A4S Multichip routing, with limitations:
  - When the `enable_a4s` parameter is false, top-level model ports are terminated with abort handlers.
  - Set `enable_a4s` parameter to true to opt into the feature.
  - The model routes A4S transactions from the `ic_dr_a4s` port to `tx_cxs_a4s` ports according to A4S LDIDs.
  - There is 1 A4S tx/rx port for each CCG up to a maximum of 32 CCGs.
  - When the remote transaction arrives at the receiving remote CMN on `rx_cxs_a4s` ports, it is routed to the GIC A4S port (`ic_rd_a4s`).
  - The A4S LDID for the CCGs can be found by reading the `ccg_RA.unit_info` register or through model parameter `print_cmn_config`.

A4S support limitations are listed in the Model Limitations section.

There is support for DSU AXU and DMC AXU interfaces:

- The DSU AXU and DMC AXU memory regions' start address are with the `dsu_periphbase` and `dmc_periphbase` parameters respectively.

The following list describes the level of support in the CMN700 model for different revision-specific features of the IP:

- r1p0, second release

#### **CXLv2.0 host-side support for CXL.mem and CXL.io protocols for Type-3 memory expansion devices**

Model supports Type-3 connections using PVBUS

#### **32 CCG or CXG gateway nodes**

Supported



**Non power-of-2 hashing of HN-Fs with  $2N * \{1, 3, \text{or } 5\}$  up to 64 HN-Fs or 128 HN-Fs with CAL**

Supported

- r2p0, third release

**Remote PCIe streaming support**

Not in scope

**1.5MB SLC support**

Supported, SLC\_SIZE=2, NUM\_WAYS=12

**90 RN-I support**

Only 40 supported (3 AXI port each)

**128 SN-F/SBSX support**

Only 80 supported

**AXID based for port aggregation across chip**

Not supported

**RNSAM support for 4 chip flat hashing configuration**

Supported

- r3p0, fourth release

**AXU port on MXPs**

Not supported

**512 RN-I requests support**

Not in scope

**16-bit REQ RSVDC support**

Width reported in info\_global register; RSVDC not in scope

**Configurable write cancel threshold in RN-I and RN-D**

Not in scope

**Remote DVM sync collapsing**

Not in scope

**CPAG MOD-3 hashing**

CPAG hashing not supported

**PCIe write streaming improvements**

Not in scope

- r3p3, fifth release

**Performance optimization guideline improvements for RN-I and RN-D**

Not in scope

**HN-P and HN-I AxID Encoding improvements**

Not in scope

**HCAL2 Discovery supported**

Feature available, but not tested for cmn700

## Model limitations

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
  - Power, clock, and interrupt signals are not supported, but RAS-related interrupts are supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch target operations are not supported.
- Non-XY routing behavior is not supported.
- The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.
- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-lite port have not been tested.
- RNSAMs external to the mesh network are not functionally supported.



Note

Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31]) of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address [MAX:12] instead of the actual address [MAX:6], due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address [MAX:6], but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.
- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.
- SLC/LCC partitioning dynamic allocation is not supported.

- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mxp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- HN-Fs with different SLC sizes in the same configuration are not supported.
- GIC communication over A4S ports is not supported.
- No support or updates for the following parameters:
  - `POR_RSVDC_STRONGNC_EN_PARAM`
  - `POR_HNSAM_CUSTOM_REGS_PARAM`
- No updates for a new bit in `CMN_HNS_CFG_CTL` to disable HNS stashing snoop (`hnf_stash_snp_dis`).
- HND-APB registers not supported.
- HN-P nodes are not supported as hashed target from the RNSAM.
- There is limited support for CXL Type-3. It only supports a single device connection (`sa_ports_cnt`).
- For `CMN700R1`, `por_hnf_cfg_ctl` follows the `CMN700R0` write mask and reset value.
- For `CMN700R1` and later, stash snooping is not supported.
- The model cannot activate both CCG APB register access traces and CMN register access traces simultaneously. Use the parameter `register_traces_for_ccg_apb_accesses` to enable CCG APB register access traces. By default, CMN register access traces are available for activation.
- MPAM features are not supported in the model, and the configuration-dependent register reset values do not match the RTL. Some of the registers (`por_hnf_mpam_idr` to `por_hnf_mpam_mbwumon_idr`) shared between security modes are not present in `MPAM_S`.
- The MPAM reset values described in `por_hnf_mpam_ns_por_hnf_mpam_idr[31:24]` default to the not-supported state. The model parameter `hnf_mpam_idr_override` can be used to override this but does not implement any of the functionality.
- The following limitations apply to System Cache Groups and Hash Target Groups:
  - The `hashed_target_grp_hnp_nodeid_reg` which is used for CCG/HN-P target IDs is not supported.
  - AXID hashing across HN-P/CCGs is not supported.
  - HTGs containing targets across multi-chips are not supported.
- `MPAM_S` `secure_register_groups_override` is not supported.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.
- For Power State Transition related flush/disable of the SLC due to `HNF_PPU_PWPR` register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.

- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.
- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` are used as the default target only when `RNSAM_STATUS.use_default_node=1`. Otherwise, when a txn is sent into the model that does not belong to any of the address regions programmed in the RNSAM, it is routed to the HN-D irrespective of what is programmed in the `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` fields.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- RSVDC StrongNC and its associated functionality is not supported.
- User-defined hashing mechanism in an SCG is not supported.
- The CXSA mode has limited support. Currently, it only supports one aggregated device.
- The model does not display any register traces.
- GenericTrace for the CMN700 Fast Model, incorrectly mentions “CMN600” in logs.
- CCG node addresses do not match RTL node addresses if not using node addresses from yml.
- The following limitations are specific to revision r2p0:
  - The model does not support the RA\_PRESENT configurable option. RAs are always present in CCG.
  - Maximum number of RN-D supported is 40.
- The following limitations apply to revisions r2p0 and r3p0:
  - No support for RWL (ReadWriteLock).
  - Maximum number of RN-I supported is 40.
  - Maximum combined number of RN-I and RN-D is 40.
  - SN-Fs on CAL4 are not supported.
  - Maximum number of SN interfaces supported is 80.
- The following limitations are specific to revision r3p0:
  - Maximum RAID of 1024 is not supported.
  - Direct Subordinate Access (DSA) CCG inbound request bypass of HN-F is not supported.
  - CXL v2.0 device support for various types is not supported or verified.
  - CXL v2.0 host support for various types is not supported or verified.

- There is no support for CPAG MOD-3 hashing.
- AXU Limitations:
  - There is no support for AXU on MXPs.
  - There is no support for APB accesses on AXU interfaces.
  - The `dsu_apb_only` and `dmc_axu_only` fields are not supported.
- RAS feature limitations:
  - Error logging and reporting functionality for CCG, HN-F, and CXHA are not supported.
  - Single-bit error injection for MTU is not supported as there is no ECC checker or register present to support it.
  - NDE response and Poison error check are not supported.
  - Flit parity and Data check errors are not supported.
  - The information that is captured as source ID, target ID, and logical ID in the ERRMISC register might not be correct or match the hardware.
  - HN-D Illegal Configuration check does not check that the access is of device type.
  - HN-D Illegal Configuration check does not check the access security mode.
  - SN-F RAS errors are treated as SBSX errors.
- A4S support limitations:
  - GIC\_DESTID input strap is not supported. Incoming transactions from remote chip are always routed to IC\_RD.
  - The model assumes the presence of 1x A4S port for GIC without regard for the actual number of a4s interfaces in `themesh_config_file` topology.
  - The model does not require user software at runtime to enable the CMN to route multichip A4S transactions between chips.
  - Limited performance testing has been performed.
  - There is no support for the use of “id\_map” file specified by CMN Configuration Integration Manual (CIM) to configure the model for reset.
- Model behavior does not reflect errata notice 2732981. The model behaves as r3p1. See the errata for details.
- For an HTG/SCG, the model does not consider the secondary region if the primary region is not valid.
- HN-D is only permitted on device port P2 in a single-MXP configuration.
- Only the last RN-F in the mesh, which is the one with the highest logical id, can be controlled by `mvp_p[0-5]_syscoreq_ctl` registers. Also, incorrectly, it can be controlled from any XP.
- HNSAM only supports two non-hashed memory regions. Memory regions programmed using `cmn_hns_sam_nonhash_cfg[1|2]_memregion2-63` registers are ignored.
- The Hierarchical hashing fields  
`HASHED_TARGET_GRP_HASH_CNTL_REG.htg_region#{index}_hier_enable_address_stripping`  
 and `HASHED_TARGET_GRP_HASH_CNTL_REG.htg_region#{index}_hier_cluster_mask` are not supported.

- The `HNSAM_DEF_HASHED_GRP_EN` yml parameter is not supported and HN-F SAM legacy mode is always enabled.
- The `HNSAM_NUM_HTG` yml parameter is not supported.
- When both `POR_CCLA_ULL_CTL.u11_to_u11_en` and `POR_CCLA_ULL_CTL.send_vd_init` bits are set then both `POR_CCLA_ULL_STATUS.tx_state` and `POR_CCLA_ULL_STATUS.rx_state` are set. The other side of the link is not consulted to set `POR_CCLA_ULL_STATUS.rx_state`.
- HN-F SAM:
  - Address masking in default hash regions in HN-F SAM is not supported.
  - Hashing in default hash regions in HN-F SAM is not supported. However, trace `SNF_HASHING_Target_SNF` displays the target SN-F as if 3SN/6SN hashing were enabled.



Note

The transaction is always routed to the SN0 nodeid programmed in the `SAM_CONTROL` register.

- Re-programming regions in HN-F SAM is not tested.
- Hashing across CCGs in HN-F SAM is not supported.
- Address compare hashed regions in HN-F SAM do not support non-power of 2 hashing. When non-power of 2 hashing is enabled, the first SN in the HTG (SN0) is used as the target.
- `HNSAM_DEF_HASHED_GRP_EN` yml parameter is not supported.
- CMN700 r1p0 supports only 8 hashed regions in HN-F SAM. CMN700 r2p0 and r3p0 support 16 hashed regions.
- Default hash regions in HN-F SAM have limited support.  
`cmn_hns_sam_cfg1_def_hashed_region` and `cmn_hns_sam_cfg2_def_hashed_region` are not supported.
- Hashing across SN-F on CAL2 in HN-F SAM is not supported.
- Remote chip addresses are incorrectly allocating a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- Model does not support APB registers.
- `POR_MTU_TAG_ADDR_CTL.memory_map_mode=3'b000` (Pass-through) is the only supported behavior. Other values for this field are not supported.
- For RN-D nodes, when software writes `SYSCOREQ`, DVM propagation gets enabled but `SYSCOACK` is not set.
- Functionality behind 'dn\_domain' yml param is abstracted away and the model forwards the DVM message to all upstream ports irrespective of 'dn\_domain' value. Reading `rnsam_status.dn_nodeid` does not reflect what `dn_domain` the node is in.

### About the `debug_force_snoop` parameter

The CMN interconnect normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream rnf ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



Note

This parameter is for debug purposes only. Do not use it instead of correctly configuring CMN.

## Iris and MTI instances for CMN700

This model has the following Iris instances:

**Table 3-914: CMN700 Iris instances**

InstanceName	ComponentName
CMN700	CMN700
CMN700.bus_slave_ocm_NS	PVBusSlave
CMN700.bus_slave_ocm_S	PVBusSlave
CMN700.cmn600_cache	PVCache
CMN700.cmn600_cache.upstream[0]	PVBusSlave
CMN700.cmn600_cache.upstream[10]	PVBusSlave
CMN700.cmn600_cache.upstream[11]	PVBusSlave

InstanceName	ComponentName
CMN700.cmn600_cache.upstream[12]	PVBusSlave
CMN700.cmn600_cache.upstream[13]	PVBusSlave
CMN700.cmn600_cache.upstream[14]	PVBusSlave
CMN700.cmn600_cache.upstream[15]	PVBusSlave
CMN700.cmn600_cache.upstream[16]	PVBusSlave
CMN700.cmn600_cache.upstream[17]	PVBusSlave
CMN700.cmn600_cache.upstream[18]	PVBusSlave
CMN700.cmn600_cache.upstream[19]	PVBusSlave
CMN700.cmn600_cache.upstream[1]	PVBusSlave
CMN700.cmn600_cache.upstream[20]	PVBusSlave
CMN700.cmn600_cache.upstream[21]	PVBusSlave
CMN700.cmn600_cache.upstream[22]	PVBusSlave
CMN700.cmn600_cache.upstream[23]	PVBusSlave
CMN700.cmn600_cache.upstream[24]	PVBusSlave
CMN700.cmn600_cache.upstream[25]	PVBusSlave
CMN700.cmn600_cache.upstream[26]	PVBusSlave
CMN700.cmn600_cache.upstream[27]	PVBusSlave
CMN700.cmn600_cache.upstream[28]	PVBusSlave
CMN700.cmn600_cache.upstream[29]	PVBusSlave
CMN700.cmn600_cache.upstream[2]	PVBusSlave
CMN700.cmn600_cache.upstream[30]	PVBusSlave
CMN700.cmn600_cache.upstream[31]	PVBusSlave
CMN700.cmn600_cache.upstream[32]	PVBusSlave
CMN700.cmn600_cache.upstream[33]	PVBusSlave
CMN700.cmn600_cache.upstream[34]	PVBusSlave
CMN700.cmn600_cache.upstream[35]	PVBusSlave
CMN700.cmn600_cache.upstream[36]	PVBusSlave
CMN700.cmn600_cache.upstream[37]	PVBusSlave
CMN700.cmn600_cache.upstream[3]	PVBusSlave
CMN700.cmn600_cache.upstream[4]	PVBusSlave
CMN700.cmn600_cache.upstream[5]	PVBusSlave
CMN700.cmn600_cache.upstream[6]	PVBusSlave
CMN700.cmn600_cache.upstream[7]	PVBusSlave
CMN700.cmn600_cache.upstream[8]	PVBusSlave
CMN700.cmn600_cache.upstream[9]	PVBusSlave
CMN700.cmn700_tag_cache	CMN_TAG_CACHE
CMN700.cmn700_tag_cache.metadata_controller0	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller0.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller1	MetaDataController



InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller1.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller10	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller10.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller100	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller100.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller101	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller101.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller102	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller102.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller103	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller103.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller104	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller104.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller105	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller105.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller106	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller106.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller107	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller107.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller108	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller108.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller109	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller109.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller11	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller11.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller110	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller110.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller111	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller111.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller112	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller112.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller113	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller113.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller114	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller114.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller115	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller115.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller116	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller116.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller117	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller117.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller118	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller118.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller119	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller119.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller12	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller12.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller120	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller120.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller121	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller121.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller122	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller122.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller123	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller123.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller124	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller124.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller125	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller125.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller126	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller126.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller127	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller127.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller13	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller13.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller14	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller14.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller15	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller15.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller16	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller16.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller17	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller17.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller18	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller18.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller19	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller19.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller2	MetaDataController

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller2.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller20	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller20.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller21	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller21.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller22	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller22.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller23	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller23.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller24	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller24.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller25	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller25.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller26	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller26.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller27	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller27.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller28	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller28.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller29	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller29.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller3	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller3.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller30	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller30.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller31	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller31.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller32	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller32.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller33	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller33.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller34	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller34.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller35	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller35.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller36	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller36.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller37	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller37.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller38	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller38.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller39	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller39.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller4	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller4.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller40	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller40.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller41	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller41.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller42	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller42.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller43	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller43.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller44	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller44.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller45	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller45.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller46	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller46.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller47	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller47.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller48	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller48.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller49	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller49.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller5	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller5.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller50	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller50.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller51	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller51.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller52	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller52.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller53	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller53.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller54	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller54.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller55	MetaDataController

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller55.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller56	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller56.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller57	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller57.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller58	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller58.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller59	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller59.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller6	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller6.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller60	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller60.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller61	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller61.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller62	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller62.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller63	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller63.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller64	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller64.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller65	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller65.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller66	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller66.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller67	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller67.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller68	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller68.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller69	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller69.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller7	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller7.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller70	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller70.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller71	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller71.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller72	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller72.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller73	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller73.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller74	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller74.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller75	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller75.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller76	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller76.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller77	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller77.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller78	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller78.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller79	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller79.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller8	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller8.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller80	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller80.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller81	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller81.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller82	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller82.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller83	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller83.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller84	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller84.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller85	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller85.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller86	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller86.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller87	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller87.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller88	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller88.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller89	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller89.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller9	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller9.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller90	MetaDataController

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller90.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller91	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller91.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller92	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller92.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller93	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller93.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller94	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller94.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller95	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller95.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller96	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller96.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller97	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller97.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller98	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller98.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller99	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller99.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.remapper0	PVBusMapper
CMN700.cmn700_tag_cache.remapper1	PVBusMapper
CMN700.cmn700_tag_cache.remapper10	PVBusMapper
CMN700.cmn700_tag_cache.remapper100	PVBusMapper
CMN700.cmn700_tag_cache.remapper101	PVBusMapper
CMN700.cmn700_tag_cache.remapper102	PVBusMapper
CMN700.cmn700_tag_cache.remapper103	PVBusMapper
CMN700.cmn700_tag_cache.remapper104	PVBusMapper
CMN700.cmn700_tag_cache.remapper105	PVBusMapper
CMN700.cmn700_tag_cache.remapper106	PVBusMapper
CMN700.cmn700_tag_cache.remapper107	PVBusMapper
CMN700.cmn700_tag_cache.remapper108	PVBusMapper
CMN700.cmn700_tag_cache.remapper109	PVBusMapper
CMN700.cmn700_tag_cache.remapper11	PVBusMapper
CMN700.cmn700_tag_cache.remapper110	PVBusMapper
CMN700.cmn700_tag_cache.remapper111	PVBusMapper
CMN700.cmn700_tag_cache.remapper112	PVBusMapper
CMN700.cmn700_tag_cache.remapper113	PVBusMapper
CMN700.cmn700_tag_cache.remapper114	PVBusMapper
CMN700.cmn700_tag_cache.remapper115	PVBusMapper



InstanceName	ComponentName
CMN700.cmn700_tag_cache.remapper116	PVBusMapper
CMN700.cmn700_tag_cache.remapper117	PVBusMapper
CMN700.cmn700_tag_cache.remapper118	PVBusMapper
CMN700.cmn700_tag_cache.remapper119	PVBusMapper
CMN700.cmn700_tag_cache.remapper12	PVBusMapper
CMN700.cmn700_tag_cache.remapper120	PVBusMapper
CMN700.cmn700_tag_cache.remapper121	PVBusMapper
CMN700.cmn700_tag_cache.remapper122	PVBusMapper
CMN700.cmn700_tag_cache.remapper123	PVBusMapper
CMN700.cmn700_tag_cache.remapper124	PVBusMapper
CMN700.cmn700_tag_cache.remapper125	PVBusMapper
CMN700.cmn700_tag_cache.remapper126	PVBusMapper
CMN700.cmn700_tag_cache.remapper127	PVBusMapper
CMN700.cmn700_tag_cache.remapper13	PVBusMapper
CMN700.cmn700_tag_cache.remapper14	PVBusMapper
CMN700.cmn700_tag_cache.remapper15	PVBusMapper
CMN700.cmn700_tag_cache.remapper16	PVBusMapper
CMN700.cmn700_tag_cache.remapper17	PVBusMapper
CMN700.cmn700_tag_cache.remapper18	PVBusMapper
CMN700.cmn700_tag_cache.remapper19	PVBusMapper
CMN700.cmn700_tag_cache.remapper2	PVBusMapper
CMN700.cmn700_tag_cache.remapper20	PVBusMapper
CMN700.cmn700_tag_cache.remapper21	PVBusMapper
CMN700.cmn700_tag_cache.remapper22	PVBusMapper
CMN700.cmn700_tag_cache.remapper23	PVBusMapper
CMN700.cmn700_tag_cache.remapper24	PVBusMapper
CMN700.cmn700_tag_cache.remapper25	PVBusMapper
CMN700.cmn700_tag_cache.remapper26	PVBusMapper
CMN700.cmn700_tag_cache.remapper27	PVBusMapper
CMN700.cmn700_tag_cache.remapper28	PVBusMapper
CMN700.cmn700_tag_cache.remapper29	PVBusMapper
CMN700.cmn700_tag_cache.remapper3	PVBusMapper
CMN700.cmn700_tag_cache.remapper30	PVBusMapper
CMN700.cmn700_tag_cache.remapper31	PVBusMapper
CMN700.cmn700_tag_cache.remapper32	PVBusMapper
CMN700.cmn700_tag_cache.remapper33	PVBusMapper
CMN700.cmn700_tag_cache.remapper34	PVBusMapper
CMN700.cmn700_tag_cache.remapper35	PVBusMapper
CMN700.cmn700_tag_cache.remapper36	PVBusMapper



InstanceName	ComponentName
CMN700.cmn700_tag_cache.remapper37	PVBusMapper
CMN700.cmn700_tag_cache.remapper38	PVBusMapper
CMN700.cmn700_tag_cache.remapper39	PVBusMapper
CMN700.cmn700_tag_cache.remapper4	PVBusMapper
CMN700.cmn700_tag_cache.remapper40	PVBusMapper
CMN700.cmn700_tag_cache.remapper41	PVBusMapper
CMN700.cmn700_tag_cache.remapper42	PVBusMapper
CMN700.cmn700_tag_cache.remapper43	PVBusMapper
CMN700.cmn700_tag_cache.remapper44	PVBusMapper
CMN700.cmn700_tag_cache.remapper45	PVBusMapper
CMN700.cmn700_tag_cache.remapper46	PVBusMapper
CMN700.cmn700_tag_cache.remapper47	PVBusMapper
CMN700.cmn700_tag_cache.remapper48	PVBusMapper
CMN700.cmn700_tag_cache.remapper49	PVBusMapper
CMN700.cmn700_tag_cache.remapper5	PVBusMapper
CMN700.cmn700_tag_cache.remapper50	PVBusMapper
CMN700.cmn700_tag_cache.remapper51	PVBusMapper
CMN700.cmn700_tag_cache.remapper52	PVBusMapper
CMN700.cmn700_tag_cache.remapper53	PVBusMapper
CMN700.cmn700_tag_cache.remapper54	PVBusMapper
CMN700.cmn700_tag_cache.remapper55	PVBusMapper
CMN700.cmn700_tag_cache.remapper56	PVBusMapper
CMN700.cmn700_tag_cache.remapper57	PVBusMapper
CMN700.cmn700_tag_cache.remapper58	PVBusMapper
CMN700.cmn700_tag_cache.remapper59	PVBusMapper
CMN700.cmn700_tag_cache.remapper6	PVBusMapper
CMN700.cmn700_tag_cache.remapper60	PVBusMapper
CMN700.cmn700_tag_cache.remapper61	PVBusMapper
CMN700.cmn700_tag_cache.remapper62	PVBusMapper
CMN700.cmn700_tag_cache.remapper63	PVBusMapper
CMN700.cmn700_tag_cache.remapper64	PVBusMapper
CMN700.cmn700_tag_cache.remapper65	PVBusMapper
CMN700.cmn700_tag_cache.remapper66	PVBusMapper
CMN700.cmn700_tag_cache.remapper67	PVBusMapper
CMN700.cmn700_tag_cache.remapper68	PVBusMapper
CMN700.cmn700_tag_cache.remapper69	PVBusMapper
CMN700.cmn700_tag_cache.remapper7	PVBusMapper
CMN700.cmn700_tag_cache.remapper70	PVBusMapper
CMN700.cmn700_tag_cache.remapper71	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.remapper72	PVBusMapper
CMN700.cmn700_tag_cache.remapper73	PVBusMapper
CMN700.cmn700_tag_cache.remapper74	PVBusMapper
CMN700.cmn700_tag_cache.remapper75	PVBusMapper
CMN700.cmn700_tag_cache.remapper76	PVBusMapper
CMN700.cmn700_tag_cache.remapper77	PVBusMapper
CMN700.cmn700_tag_cache.remapper78	PVBusMapper
CMN700.cmn700_tag_cache.remapper79	PVBusMapper
CMN700.cmn700_tag_cache.remapper8	PVBusMapper
CMN700.cmn700_tag_cache.remapper80	PVBusMapper
CMN700.cmn700_tag_cache.remapper81	PVBusMapper
CMN700.cmn700_tag_cache.remapper82	PVBusMapper
CMN700.cmn700_tag_cache.remapper83	PVBusMapper
CMN700.cmn700_tag_cache.remapper84	PVBusMapper
CMN700.cmn700_tag_cache.remapper85	PVBusMapper
CMN700.cmn700_tag_cache.remapper86	PVBusMapper
CMN700.cmn700_tag_cache.remapper87	PVBusMapper
CMN700.cmn700_tag_cache.remapper88	PVBusMapper
CMN700.cmn700_tag_cache.remapper89	PVBusMapper
CMN700.cmn700_tag_cache.remapper9	PVBusMapper
CMN700.cmn700_tag_cache.remapper90	PVBusMapper
CMN700.cmn700_tag_cache.remapper91	PVBusMapper
CMN700.cmn700_tag_cache.remapper92	PVBusMapper
CMN700.cmn700_tag_cache.remapper93	PVBusMapper
CMN700.cmn700_tag_cache.remapper94	PVBusMapper
CMN700.cmn700_tag_cache.remapper95	PVBusMapper
CMN700.cmn700_tag_cache.remapper96	PVBusMapper
CMN700.cmn700_tag_cache.remapper97	PVBusMapper
CMN700.cmn700_tag_cache.remapper98	PVBusMapper
CMN700.cmn700_tag_cache.remapper99	PVBusMapper
CMN700.dmc_axu_mapper	PVBusMapper
CMN700.dsu_axu_mapper	PVBusMapper
CMN700.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN700.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN700.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN700.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN700.hni_exclusive_monitor1	PVBusExclusiveMonitor
CMN700.hni_exclusive_monitor1.bus_mapper	PVBusMapper
CMN700.hni_exclusive_monitor2	PVBusExclusiveMonitor

InstanceName	ComponentName
CMN700.hni_exclusive_monitor2.bus_mapper	PVBusMapper
CMN700.hni_exclusive_monitor3	PVBusExclusiveMonitor
CMN700.hni_exclusive_monitor3.bus_mapper	PVBusMapper
CMN700.ocm_decoder	PVBusMapper
CMN700.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN700.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN700.snf_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-915: CMN700 MTI instances**

InstanceName	ComponentName
CMN700	CMN700
CMN700.bus_slave_ocm_NS	PVBusSlave
CMN700.bus_slave_ocm_S	PVBusSlave
CMN700.cmn600_cache	CMN600Cache
CMN700.cmn600_cache.upstream[0]	PVBusSlave
CMN700.cmn600_cache.upstream[10]	PVBusSlave
CMN700.cmn600_cache.upstream[11]	PVBusSlave
CMN700.cmn600_cache.upstream[12]	PVBusSlave
CMN700.cmn600_cache.upstream[13]	PVBusSlave
CMN700.cmn600_cache.upstream[14]	PVBusSlave
CMN700.cmn600_cache.upstream[15]	PVBusSlave
CMN700.cmn600_cache.upstream[16]	PVBusSlave
CMN700.cmn600_cache.upstream[17]	PVBusSlave
CMN700.cmn600_cache.upstream[18]	PVBusSlave
CMN700.cmn600_cache.upstream[19]	PVBusSlave
CMN700.cmn600_cache.upstream[1]	PVBusSlave
CMN700.cmn600_cache.upstream[20]	PVBusSlave
CMN700.cmn600_cache.upstream[21]	PVBusSlave
CMN700.cmn600_cache.upstream[22]	PVBusSlave
CMN700.cmn600_cache.upstream[23]	PVBusSlave
CMN700.cmn600_cache.upstream[24]	PVBusSlave
CMN700.cmn600_cache.upstream[25]	PVBusSlave
CMN700.cmn600_cache.upstream[26]	PVBusSlave
CMN700.cmn600_cache.upstream[27]	PVBusSlave
CMN700.cmn600_cache.upstream[28]	PVBusSlave
CMN700.cmn600_cache.upstream[29]	PVBusSlave
CMN700.cmn600_cache.upstream[2]	PVBusSlave
CMN700.cmn600_cache.upstream[30]	PVBusSlave

InstanceName	ComponentName
CMN700.cmn600_cache.upstream[31]	PVBusSlave
CMN700.cmn600_cache.upstream[32]	PVBusSlave
CMN700.cmn600_cache.upstream[33]	PVBusSlave
CMN700.cmn600_cache.upstream[34]	PVBusSlave
CMN700.cmn600_cache.upstream[35]	PVBusSlave
CMN700.cmn600_cache.upstream[36]	PVBusSlave
CMN700.cmn600_cache.upstream[37]	PVBusSlave
CMN700.cmn600_cache.upstream[3]	PVBusSlave
CMN700.cmn600_cache.upstream[4]	PVBusSlave
CMN700.cmn600_cache.upstream[5]	PVBusSlave
CMN700.cmn600_cache.upstream[6]	PVBusSlave
CMN700.cmn600_cache.upstream[7]	PVBusSlave
CMN700.cmn600_cache.upstream[8]	PVBusSlave
CMN700.cmn600_cache.upstream[9]	PVBusSlave
CMN700.cmn700_tag_cache	CMNTAGCACHECADI
CMN700.cmn700_tag_cache.metadata_controller0	MetadataController
CMN700.cmn700_tag_cache.metadata_controller0.MetadataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller1	MetadataController
CMN700.cmn700_tag_cache.metadata_controller1.MetadataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller10	MetadataController
CMN700.cmn700_tag_cache.metadata_controller10.MetadataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller100	MetadataController
CMN700.cmn700_tag_cache.metadata_controller100.MetadataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller101	MetadataController
CMN700.cmn700_tag_cache.metadata_controller101.MetadataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller102	MetadataController
CMN700.cmn700_tag_cache.metadata_controller102.MetadataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller103	MetadataController
CMN700.cmn700_tag_cache.metadata_controller103.MetadataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller104	MetadataController
CMN700.cmn700_tag_cache.metadata_controller104.MetadataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller105	MetadataController
CMN700.cmn700_tag_cache.metadata_controller105.MetadataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller106	MetadataController
CMN700.cmn700_tag_cache.metadata_controller106.MetadataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller107	MetadataController
CMN700.cmn700_tag_cache.metadata_controller107.MetadataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller108	MetadataController
CMN700.cmn700_tag_cache.metadata_controller108.MetadataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller109	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller109.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller11	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller11.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller110	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller110.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller111	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller111.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller112	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller112.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller113	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller113.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller114	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller114.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller115	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller115.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller116	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller116.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller117	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller117.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller118	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller118.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller119	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller119.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller12	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller12.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller120	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller120.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller121	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller121.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller122	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller122.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller123	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller123.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller124	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller124.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller125	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller125.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller126	MetaDataController

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller126.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller127	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller127.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller13	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller13.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller14	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller14.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller15	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller15.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller16	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller16.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller17	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller17.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller18	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller18.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller19	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller19.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller2	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller2.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller20	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller20.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller21	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller21.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller22	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller22.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller23	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller23.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller24	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller24.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller25	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller25.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller26	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller26.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller27	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller27.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller28	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller28.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller29	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller29.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller3	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller3.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller30	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller30.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller31	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller31.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller32	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller32.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller33	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller33.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller34	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller34.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller35	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller35.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller36	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller36.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller37	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller37.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller38	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller38.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller39	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller39.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller4	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller4.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller40	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller40.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller41	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller41.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller42	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller42.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller43	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller43.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller44	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller44.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller45	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller45.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller46	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller46.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller47	MetaDataController



InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller47.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller48	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller48.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller49	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller49.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller5	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller5.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller50	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller50.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller51	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller51.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller52	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller52.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller53	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller53.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller54	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller54.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller55	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller55.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller56	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller56.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller57	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller57.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller58	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller58.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller59	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller59.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller6	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller6.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller60	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller60.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller61	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller61.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller62	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller62.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller63	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller63.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller64	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller64.MetaDataMapper	PVBusMapper



InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller65	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller65.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller66	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller66.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller67	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller67.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller68	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller68.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller69	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller69.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller7	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller7.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller70	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller70.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller71	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller71.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller72	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller72.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller73	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller73.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller74	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller74.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller75	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller75.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller76	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller76.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller77	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller77.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller78	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller78.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller79	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller79.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller8	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller8.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller80	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller80.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller81	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller81.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller82	MetaDataController

InstanceName	ComponentName
CMN700.cmn700_tag_cache.metadata_controller82.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller83	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller83.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller84	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller84.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller85	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller85.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller86	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller86.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller87	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller87.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller88	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller88.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller89	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller89.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller9	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller9.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller90	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller90.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller91	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller91.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller92	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller92.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller93	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller93.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller94	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller94.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller95	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller95.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller96	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller96.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller97	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller97.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller98	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller98.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controller99	MetaDataController
CMN700.cmn700_tag_cache.metadata_controller99.MetaDataMapper	PVBusMapper
CMN700.cmn700_tag_cache.remapper0	PVBusMapper
CMN700.cmn700_tag_cache.remapper1	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.remapper10	PVBusMapper
CMN700.cmn700_tag_cache.remapper100	PVBusMapper
CMN700.cmn700_tag_cache.remapper101	PVBusMapper
CMN700.cmn700_tag_cache.remapper102	PVBusMapper
CMN700.cmn700_tag_cache.remapper103	PVBusMapper
CMN700.cmn700_tag_cache.remapper104	PVBusMapper
CMN700.cmn700_tag_cache.remapper105	PVBusMapper
CMN700.cmn700_tag_cache.remapper106	PVBusMapper
CMN700.cmn700_tag_cache.remapper107	PVBusMapper
CMN700.cmn700_tag_cache.remapper108	PVBusMapper
CMN700.cmn700_tag_cache.remapper109	PVBusMapper
CMN700.cmn700_tag_cache.remapper11	PVBusMapper
CMN700.cmn700_tag_cache.remapper110	PVBusMapper
CMN700.cmn700_tag_cache.remapper111	PVBusMapper
CMN700.cmn700_tag_cache.remapper112	PVBusMapper
CMN700.cmn700_tag_cache.remapper113	PVBusMapper
CMN700.cmn700_tag_cache.remapper114	PVBusMapper
CMN700.cmn700_tag_cache.remapper115	PVBusMapper
CMN700.cmn700_tag_cache.remapper116	PVBusMapper
CMN700.cmn700_tag_cache.remapper117	PVBusMapper
CMN700.cmn700_tag_cache.remapper118	PVBusMapper
CMN700.cmn700_tag_cache.remapper119	PVBusMapper
CMN700.cmn700_tag_cache.remapper12	PVBusMapper
CMN700.cmn700_tag_cache.remapper120	PVBusMapper
CMN700.cmn700_tag_cache.remapper121	PVBusMapper
CMN700.cmn700_tag_cache.remapper122	PVBusMapper
CMN700.cmn700_tag_cache.remapper123	PVBusMapper
CMN700.cmn700_tag_cache.remapper124	PVBusMapper
CMN700.cmn700_tag_cache.remapper125	PVBusMapper
CMN700.cmn700_tag_cache.remapper126	PVBusMapper
CMN700.cmn700_tag_cache.remapper127	PVBusMapper
CMN700.cmn700_tag_cache.remapper13	PVBusMapper
CMN700.cmn700_tag_cache.remapper14	PVBusMapper
CMN700.cmn700_tag_cache.remapper15	PVBusMapper
CMN700.cmn700_tag_cache.remapper16	PVBusMapper
CMN700.cmn700_tag_cache.remapper17	PVBusMapper
CMN700.cmn700_tag_cache.remapper18	PVBusMapper
CMN700.cmn700_tag_cache.remapper19	PVBusMapper
CMN700.cmn700_tag_cache.remapper2	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.remapper20	PVBusMapper
CMN700.cmn700_tag_cache.remapper21	PVBusMapper
CMN700.cmn700_tag_cache.remapper22	PVBusMapper
CMN700.cmn700_tag_cache.remapper23	PVBusMapper
CMN700.cmn700_tag_cache.remapper24	PVBusMapper
CMN700.cmn700_tag_cache.remapper25	PVBusMapper
CMN700.cmn700_tag_cache.remapper26	PVBusMapper
CMN700.cmn700_tag_cache.remapper27	PVBusMapper
CMN700.cmn700_tag_cache.remapper28	PVBusMapper
CMN700.cmn700_tag_cache.remapper29	PVBusMapper
CMN700.cmn700_tag_cache.remapper3	PVBusMapper
CMN700.cmn700_tag_cache.remapper30	PVBusMapper
CMN700.cmn700_tag_cache.remapper31	PVBusMapper
CMN700.cmn700_tag_cache.remapper32	PVBusMapper
CMN700.cmn700_tag_cache.remapper33	PVBusMapper
CMN700.cmn700_tag_cache.remapper34	PVBusMapper
CMN700.cmn700_tag_cache.remapper35	PVBusMapper
CMN700.cmn700_tag_cache.remapper36	PVBusMapper
CMN700.cmn700_tag_cache.remapper37	PVBusMapper
CMN700.cmn700_tag_cache.remapper38	PVBusMapper
CMN700.cmn700_tag_cache.remapper39	PVBusMapper
CMN700.cmn700_tag_cache.remapper4	PVBusMapper
CMN700.cmn700_tag_cache.remapper40	PVBusMapper
CMN700.cmn700_tag_cache.remapper41	PVBusMapper
CMN700.cmn700_tag_cache.remapper42	PVBusMapper
CMN700.cmn700_tag_cache.remapper43	PVBusMapper
CMN700.cmn700_tag_cache.remapper44	PVBusMapper
CMN700.cmn700_tag_cache.remapper45	PVBusMapper
CMN700.cmn700_tag_cache.remapper46	PVBusMapper
CMN700.cmn700_tag_cache.remapper47	PVBusMapper
CMN700.cmn700_tag_cache.remapper48	PVBusMapper
CMN700.cmn700_tag_cache.remapper49	PVBusMapper
CMN700.cmn700_tag_cache.remapper5	PVBusMapper
CMN700.cmn700_tag_cache.remapper50	PVBusMapper
CMN700.cmn700_tag_cache.remapper51	PVBusMapper
CMN700.cmn700_tag_cache.remapper52	PVBusMapper
CMN700.cmn700_tag_cache.remapper53	PVBusMapper
CMN700.cmn700_tag_cache.remapper54	PVBusMapper
CMN700.cmn700_tag_cache.remapper55	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.remapper56	PVBusMapper
CMN700.cmn700_tag_cache.remapper57	PVBusMapper
CMN700.cmn700_tag_cache.remapper58	PVBusMapper
CMN700.cmn700_tag_cache.remapper59	PVBusMapper
CMN700.cmn700_tag_cache.remapper6	PVBusMapper
CMN700.cmn700_tag_cache.remapper60	PVBusMapper
CMN700.cmn700_tag_cache.remapper61	PVBusMapper
CMN700.cmn700_tag_cache.remapper62	PVBusMapper
CMN700.cmn700_tag_cache.remapper63	PVBusMapper
CMN700.cmn700_tag_cache.remapper64	PVBusMapper
CMN700.cmn700_tag_cache.remapper65	PVBusMapper
CMN700.cmn700_tag_cache.remapper66	PVBusMapper
CMN700.cmn700_tag_cache.remapper67	PVBusMapper
CMN700.cmn700_tag_cache.remapper68	PVBusMapper
CMN700.cmn700_tag_cache.remapper69	PVBusMapper
CMN700.cmn700_tag_cache.remapper7	PVBusMapper
CMN700.cmn700_tag_cache.remapper70	PVBusMapper
CMN700.cmn700_tag_cache.remapper71	PVBusMapper
CMN700.cmn700_tag_cache.remapper72	PVBusMapper
CMN700.cmn700_tag_cache.remapper73	PVBusMapper
CMN700.cmn700_tag_cache.remapper74	PVBusMapper
CMN700.cmn700_tag_cache.remapper75	PVBusMapper
CMN700.cmn700_tag_cache.remapper76	PVBusMapper
CMN700.cmn700_tag_cache.remapper77	PVBusMapper
CMN700.cmn700_tag_cache.remapper78	PVBusMapper
CMN700.cmn700_tag_cache.remapper79	PVBusMapper
CMN700.cmn700_tag_cache.remapper8	PVBusMapper
CMN700.cmn700_tag_cache.remapper80	PVBusMapper
CMN700.cmn700_tag_cache.remapper81	PVBusMapper
CMN700.cmn700_tag_cache.remapper82	PVBusMapper
CMN700.cmn700_tag_cache.remapper83	PVBusMapper
CMN700.cmn700_tag_cache.remapper84	PVBusMapper
CMN700.cmn700_tag_cache.remapper85	PVBusMapper
CMN700.cmn700_tag_cache.remapper86	PVBusMapper
CMN700.cmn700_tag_cache.remapper87	PVBusMapper
CMN700.cmn700_tag_cache.remapper88	PVBusMapper
CMN700.cmn700_tag_cache.remapper89	PVBusMapper
CMN700.cmn700_tag_cache.remapper9	PVBusMapper
CMN700.cmn700_tag_cache.remapper90	PVBusMapper

InstanceName	ComponentName
CMN700.cmn700_tag_cache.remapper91	PVBusMapper
CMN700.cmn700_tag_cache.remapper92	PVBusMapper
CMN700.cmn700_tag_cache.remapper93	PVBusMapper
CMN700.cmn700_tag_cache.remapper94	PVBusMapper
CMN700.cmn700_tag_cache.remapper95	PVBusMapper
CMN700.cmn700_tag_cache.remapper96	PVBusMapper
CMN700.cmn700_tag_cache.remapper97	PVBusMapper
CMN700.cmn700_tag_cache.remapper98	PVBusMapper
CMN700.cmn700_tag_cache.remapper99	PVBusMapper
CMN700.dmc_axu_mapper	PVBusMapper
CMN700.dsu_axu_mapper	PVBusMapper
CMN700.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN700.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN700.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN700.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN700.hni_exclusive_monitor1	PVBusExclusiveMonitor
CMN700.hni_exclusive_monitor1.bus_mapper	PVBusMapper
CMN700.hni_exclusive_monitor2	PVBusExclusiveMonitor
CMN700.hni_exclusive_monitor2.bus_mapper	PVBusMapper
CMN700.hni_exclusive_monitor3	PVBusExclusiveMonitor
CMN700.hni_exclusive_monitor3.bus_mapper	PVBusMapper
CMN700.ocm_decoder	PVBusMapper
CMN700.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN700.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN700.snf_mapper	PVBusMapper

## Ports for CMN700

**Table 3-916: Ports**

Name	Protocol	Type	Description
event_clusters	Signal	Peer	CPU event communication signal from the clusters.
event_downstream_link_signal[32]	Signal	Master	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.
event_upstream_link_signal[32]	Signal	Slave	Event from the Hub towards the CMN
ic_dr_a4s	PVBus	Slave	Interrupt Controller Distributor-to-Remote AXI4Stream port.
ic_rd_a4s	PVBus	Master	Interrupt Controller Remote-to-Distributor AXI4Stream port.
intreqerrns_irq_out	Signal	Master	Interrupt signal
intreqerrs_irq_out	Signal	Master	Interrupt signal

Name	Protocol	Type	Description
intreqfaultns_irq_out	Signal	Master	Interrupt signal
intreqfaults_irq_out	Signal	Master	Interrupt signal
pvbus_m_cml	PVBus	Master	CML downstream ports
pvbus_m_cml_cfg	PVBus	Master	CML downstream hub configuration port
pvbus_m_cxs[32]	PVBus	Master	CXS downstream ports
pvbus_m_dmc_axu[16]	PVBus	Master	DMC AXU ports
pvbus_m_dsu_axu[256]	PVBus	Master	DSU AXU ports
pvbus_m_hni[32]	PVBus	Master	HNI downstream ports.
pvbus_m_snf[128]	PVBus	Master	SNF downstream port.
pvbus_s_apb	PVBus	Slave	APB interface port.
pvbus_s_ccg_apb[32]	PVBus	Slave	CCG APB interface port.
pvbus_s_cml	PVBus	Slave	CML upstream ports
pvbus_s_rnf[256]	PVBus	Slave	RNF upstream ports.
pvbus_s_rni[270]	PVBus	Slave	RNI upstream ports. NOTE the upper 150 ports are only used in r2/r3.
reset_in	Signal	Slave	Reset signal.
rnf_sci_s[256]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in[256]	Signal	Slave	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s[120]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in[270]	Signal	Slave	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected. NOTE the upper 150 ports are only used in r2/r3.
rx_cxs_a4s[32]	PVBus	Slave	Receive channel of A4S packets from a remote CMN.
tx_cxs_a4s[32]	PVBus	Master	Transmit channel of A4S packets to a remote CMN.

## Parameters for CMN700

### a4s\_logicalid

#### Type

string

#### Default value

0

A4S ID mapping of the GIC destination component connected through a CCG port.

Specify the `CCG_NODE_ID` and the destination A4S Logical ID of the GIC component connected by using a decimal number format like:

```
<CCG_NODEID0>=<A4S_LID0>,<CCG_NODEID1>=<A4S_LID1>
```

For example for CCG Node ID 54 with A4S ID 12 - 54=12.

All of the CCG nodes must be specified.

The parameter is only valid when the `enable_a4s` is also enabled. The default behavior without this parameter is to automatically assign an incrementing A4S ID

### **acchannelen\_rnf**

#### **Type**

string

#### **Default value**

"0"



Note

DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each `rnf` port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`

### **acchannelen\_rni**

#### **Type**

string

#### **Default value**

"0"



Note

DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`



**cache\_state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Model the cache state.

**cmn700\_tag\_cache.metadata\_controller.init\_value****Type**

int

**Default value**

0xd

**Description**

Initialize metadata memory with this value. If one of init\_values\_json or init\_values\_json\_file is specified, this value applies only to any metadata not specified in the JSON.

**cmn700\_tag\_cache.metadata\_controller.init\_values\_json****Type**

string

**Default value**

""

**Description**

A JSON value describing initial metadata values. Mutually exclusive with init\_values\_json\_file. The format is as follows: {"regions": [{ "begin": 0x0, "end": 0x10000, "mte\_tag": 0xa }, {"begin": 0x20000, "end": 0x50000, "mte\_tag": 0xc }]}.

**cmn700\_tag\_cache.metadata\_controller.init\_values\_json\_file****Type**

string

**Default value**

""

**Description**

Path to a JSON file with initial metadata values. Mutually exclusive with init\_values\_json. The format is as follows: {"regions": [{ "begin": 0x0, "end": 0x10000, "mte\_tag": 0xa }, {"begin": 0x20000, "end": 0x50000, "mte\_tag": 0xc }]}.

**cmn700\_tag\_cache.metadata\_controller.is\_enabled****Type**

bool

**Default value**

0x0

**Description**

If false, disables the MetaData controller functionality, and makes the component invisible to passing transactions.

**cmn700\_tag\_cache.metadata\_controller.mte\_tag\_carveout\_json****Type**

string

**Default value**

""

**Description**

JSON string that specifies the PA range of the tag carveout regions and the beginning of the PA range for which they provide tag storage. If `pa_regions_with_metadata_storage` defines which regions can have metadata, the tag carveout regions cannot overlap them, and each tagged region must be entirely covered by one of them. The block size must be  $\geq 64$  bytes and a power of 2, defaulting to 4KiB. The maximum block size supported is 4KiB. The carveout region size must be  $\geq 4$ KiB and a power of 2, and determines the size of the corresponding tagged region. { "regions": [ { "begin": 0x0, "tag\_carveout\_region": [0xfffff00000, 0xfffff00fff]}, { "begin": 0x20000, "tag\_carveout\_region": [0xfffff01000, 0xfffff01fff], "block\_size": 0x100}, { "begin": 0x100000, "tag\_carveout\_region": [0xfffff08000, 0xfffff0Bfff], "block\_size": 0x2000}]}.

**cmn700\_tag\_cache.metadata\_controller.mte\_tag\_carveout\_json\_file****Type**

string

**Default value**

""

**Description**

Path to a JSON file that specifies the PA range of the tag carveout regions with the same format as `mte_tag_carveout_json`. Only one of `mte_tag_carveout_json` and `mte_tag_carveout_json_file` can be used. .

**cmn700\_tag\_cache.metadata\_controller.mte\_tag\_carveout\_tag\_order****Type**

string

**Default value**

little-endian

**Description**

Order of the tags within the MTE tag carveout blocks. This can be little-endian (same order as the corresponding tagged data) or big-endian (reverse order). The parameter accepts both '-' and '\_', so 'little-endian', 'big-endian', 'little\_endian' and 'big\_endian' are all valid. .

**cmn700\_tag\_cache.metadata\_controller.pa\_regions\_with\_metadata\_storage****Type**

string

**Default value**

""

**Description**

Specify the address region where the metadata storage is available for each PAS in a JSON format. If the PAS does not have a region specified, the PAS has metadata storage for all of the space. The regions are defined by begin and end\_incl addresses. Example: { "ns": [0xa0000000, 0xa0000fff], "s": [0xb0000000, 0xb0000fff], "rl": [0xc0000000, 0xc0000fff], "rt": [0xd0000000, 0xd0000fff] } ns: non-secure, s: secure, rl: realm, rt: root.

**debug\_force\_snoop****Type**

bool

**Default value**

false

The CMN700 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.



This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

**disable\_CML\_port****Type**

bool

**Default value**

0x0

**Description**

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

**dmc\_periphbase****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Value for DMC\_PERIPHBASE.

**dsu\_periphbase****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Value for DSU\_PERIPHBASE.

**enable\_a4s****Type**

bool

**Default value**

0x0

**Description**

Enables A4S ports for GIC multi-chip routing.

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enable PVBusLoggers for the downstream ports in the CMN model.

**enable\_ras****Type**

bool

**Default value**

0x0

**Description**

Enables RAS. There is an impact on performance when RAS is enabled.

**`enable_rnsam_to_hnf_wider_hash`****Type**

bool

**Default value**

false

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

**`force_rnsam_internal`****Type**

bool

**Default value**

0x1

**Description**

Force all RNSAMs to be internal independently of the mesh topology.

**`hnf_mpam_idr_override`****Type**

uint64\_t

**Default value**

0

Set to override `hnf_mpam_idr[31:24]` value. Bit[28] is Reserved and is ignored.

**`mesh_config_file`****Type**

string

**Default value**

""

**Description**

Name of a file containing mesh placement of CMN700 components.

**periphbase**

**Type**  
uint64\_t

**Default value**  
0x20000000

Value for PERIPHBASE. Bits [27:0] are treated 0

**print\_cmn\_ccix\_config**

**Type**  
bool

**Default value**  
0x0

**Description**  
Print information about the CCIX configuration.

**print\_cmn\_config**

**Type**  
bool

**Default value**  
0x0

**Description**  
Print the mesh topology and children pointers acquired from the YML file.

**register\_traces\_for\_ccg\_apb\_accesses**

**Type**  
bool

**Default value**  
false



Will be removed when enhancement SDDKW-74284 is done.

---

Intended for use with trace plugins.

**true**  
registers traces to CCG register accesses through CCG APB interface.

**false**  
registers traces to CMN register accesses through all other interfaces (eg RN nodes).

**revision****Type**

string

**Default value**

"r0p0"

Component revision.

Currently supports r0p0, r1p0, r2p0, r3p0, r3p3.

**rnf\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

**rni\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

**show\_banner****Type**

uint64\_t

**Default value**

2

Show component banner:

**0**

supress entire banner

**1**

suppress config file

**2+**

show full banner

**skip\_cmn\_config\_check****Type**

bool

**Default value**

0x0

**Description**

Skip any topology configuration checks. The maximum number of devices per type not verified.

**use\_yaml\_periphbase****Type**

bool

**Default value**

false

Use yaml param CFGM\_PERIPHBASE\_PARAM to specify periphbase address.

If false, model parameter periphbase will be used.

**yaml\_has\_node\_addresses****Type**

bool

**Default value**

0x0

**Description**

Does the top-level YAML file describe node-addresses ?.



### 3.10.20 CMN\_S3

CMN\_S3 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-917: IP revisions support**

Revision	Quality level
r0p0	Preliminary support
r0p1	Preliminary support
r2p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Changes in 11.29.19

Ports added:

- `pvbus_m_dmc_axu`
- `pvbus_m_dsu_axu`
- `pvbus_m_mxp_axu`
- `pvbus_s_cxs`

Ports removed:

- `pvbus_m_cml`
- `pvbus_m_cml_cfg`
- `pvbus_s_cml`
- `pvc2c_m`
- `pvc2c_s`

Parameters added:

- `dmc_periphbase`
- `dsu_periphbase`
- `mxp_axu_periphbase`

Parameters removed:

- `disable_CML_port`

#### About CMN\_S3

- Major IP revisions (rX) are modeled and are controlled by the `version` parameter in the topology file.

- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of CHI nodes. Set it to the name of the yml configuration file emitted by Socrates. You must use r1p7-06 version of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.

The following functionality is supported:

- CHI Issue F Realm Management Extension (RME) with `LEGACY_TZ_EN`.
  - `LEGACY_TZ_EN` is controlled by a model parameter.
    - RCR registers are **RAZ/WI**.
    - Does not prevent REALM/ROOT transactions from flowing through interconnect.
- HNS device isolation.
- Hybrid CAL3 for discovery only.
- CPA groups including multi CPA groups.
- Large systems can be built by connecting multiple CMNs (Coherent Mesh Networks) using their CCG components. These are referred to as “multi-chip” systems. In real hardware the CCG components communicate using the CXS protocol. The CMN Fast Models do not implement CXS, but PVBUS can be used to represent CCG connections.
  - `pvbus_m_cxs` and `pvbus_s_cxs` ports represent the CCG ports
  - Connecting `cmnA.pvbus_m_cxs[a]` to `cmnB.pvbus_s_cxs[b]`, represents connecting `cmnA.CCGa` to `cmnB.CCGB`; the reverse connection should also be made: `cmnB.pvbus_m_cxs[b]` to `cmnA.pvbus_s_cxs[a]`.
  - Caching functionality (and consequently snooping) is not supported in multichip platforms.
  - DVM's work correctly only in systems where all CMN's are connected to all other CMN's in the system.
  - `pvbus_m_cxs` ports can also be used to connect to CXL Type-3 devices.
  - No support for the CMLHub to model multi-chip systems.
- MXP AXU interfaces are supported.
  - The base address for accesses on these interfaces must be specified through the `mxp_axu_periphbase` parameter.

## Model limitations



Issues listed in this section have identifiers of the form [SDDKW-x]. These are Arm internal references only.

- [SDDKW-82504] CMN700 Fast Model limitations apply to CMN\_S3 model as well, in addition to those listed below. See [CMN700](#) for limitations.
- RAS functionality is not supported.
- Features not supported:
  - CHI C2C is not supported.
  - Direct Subordinate Access (DSA)
  - MTSX
  - Port to Port Forwarding for CML SMP
  - Memory Protection Engine (MPE) for CXL-Type Host
- Out of scope features:
  - Datasource handling
- [SDDKW-81128] HNS device isolation:
  - Hashed target groups are checked for disabled nodes at the time the RNSAM is programmed and ready (`rnsam_status.nstall_req=1'b1`). Disabling HNS devices after RNSAM programming is a no-op.
  - Having disabled HNS nodes in a hash target group is a non-fatal error. Model displays an error message and behaves the same as if the node was enabled.
  - The default scenario is not supported, i.e. disabling only one device of a CAL2 pair in a hash target group triggers a non-fatal error message for that specific device of the CAL2 pair. The error message does not affect the other device behind CAL2.
  - `por_mxp_child_pointer_0-31`'s bit 30 does not indicate if the node is isolated or not. This is applicable to `r2p0` and above.
- [SDDKW-82529] OCM does not support RL/RT PAS
- [SDDKW-79584] `por_cfgm_periph_id_0_periph_id_1` has the value `0x000000b40000003e` which may not match RTL
- [SDDKW-76061] Model does not support global secure overrides for root registers.
- [SDDKW-76516] Model does not support RCR (root override register) for the following nodes:
  - DT
  - DN
  - MTU
  - APB
- [SDDKW-80175] DSU HNI region in RNSAM is not supported.
- [SDDKW-79842] Increased maximum HN-I limit to 48. Maximum tested is 46.
- [SDDKW-85869] `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` is used as the default target only when `RNSAM_STATUS.use_default_node=1`. Otherwise, when a `txn` is sent into the model which does not belong to any of the address regions programmed in the RNSAM, it is routed to the HND irrespective of what is programmed in the `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` fields

- [SDDKW-85945]  
HASHED\_TARGET\_GRP\_HASH\_CNTL\_REG0-31.htg\_region#{index}\_hier\_enable\_address\_stripping and HASHED\_TARGET\_GRP\_HASH\_CNTL\_REG0-31.htg\_region#{index}\_hier\_cluster\_mask are not supported.
- Hybrid CAL
  - [SDDKW-90932] CAL2 has not been tested.
  - [SDDKW-90932] CAL3 functionality beyond discovery has not been tested.
- [SDDKW-85839] Register bitfields with W1S are treated as RW.
- HNP on CAL4 are not supported.
- HNS on CAL4 are not tested.
- [SDDKW-91300] HCAL3 is not supported.
- NUM\_EXCL\_CHIPS and MAX\_EXCL\_ALL\_CHIPS parameter values are not reflected in register por\_info\_global\_1.
- LCNSAM\_NUM\_HTG, LCNSAM\_NUM\_NONHASHGROUP parameter values are not reflected in register cmn\_hns\_unit\_info\_1.
- NUM\_VMF parameter value is not reflected in por\_dn\_build\_info.
- AXU
  - APB accesses on MXP AXU interfaces are not supported.
  - The mxp\_axu\_only field is not supported.

### Iris and MTI instances for CMN\_S3

This model has the following Iris instances:

**Table 3-918: CMN\_S3 Iris instances**

InstanceName	ComponentName
CMN_S3	CMN700
CMN_S3.bus_slave_ocm_NS	PVBusSlave
CMN_S3.bus_slave_ocm_S	PVBusSlave
CMN_S3.cmn600_cache	PVCache
CMN_S3.cmn600_cache.upstream[0]	PVBusSlave
CMN_S3.cmn600_cache.upstream[100]	PVBusSlave
CMN_S3.cmn600_cache.upstream[101]	PVBusSlave
CMN_S3.cmn600_cache.upstream[102]	PVBusSlave
CMN_S3.cmn600_cache.upstream[103]	PVBusSlave
CMN_S3.cmn600_cache.upstream[104]	PVBusSlave
CMN_S3.cmn600_cache.upstream[105]	PVBusSlave
CMN_S3.cmn600_cache.upstream[106]	PVBusSlave
CMN_S3.cmn600_cache.upstream[107]	PVBusSlave
CMN_S3.cmn600_cache.upstream[108]	PVBusSlave
CMN_S3.cmn600_cache.upstream[109]	PVBusSlave

InstanceName	ComponentName
CMN_S3.cmn600_cache.upstream[10]	PVBusSlave
CMN_S3.cmn600_cache.upstream[110]	PVBusSlave
CMN_S3.cmn600_cache.upstream[111]	PVBusSlave
CMN_S3.cmn600_cache.upstream[112]	PVBusSlave
CMN_S3.cmn600_cache.upstream[113]	PVBusSlave
CMN_S3.cmn600_cache.upstream[114]	PVBusSlave
CMN_S3.cmn600_cache.upstream[115]	PVBusSlave
CMN_S3.cmn600_cache.upstream[116]	PVBusSlave
CMN_S3.cmn600_cache.upstream[11]	PVBusSlave
CMN_S3.cmn600_cache.upstream[12]	PVBusSlave
CMN_S3.cmn600_cache.upstream[13]	PVBusSlave
CMN_S3.cmn600_cache.upstream[14]	PVBusSlave
CMN_S3.cmn600_cache.upstream[15]	PVBusSlave
CMN_S3.cmn600_cache.upstream[16]	PVBusSlave
CMN_S3.cmn600_cache.upstream[17]	PVBusSlave
CMN_S3.cmn600_cache.upstream[18]	PVBusSlave
CMN_S3.cmn600_cache.upstream[19]	PVBusSlave
CMN_S3.cmn600_cache.upstream[1]	PVBusSlave
CMN_S3.cmn600_cache.upstream[20]	PVBusSlave
CMN_S3.cmn600_cache.upstream[21]	PVBusSlave
CMN_S3.cmn600_cache.upstream[22]	PVBusSlave
CMN_S3.cmn600_cache.upstream[23]	PVBusSlave
CMN_S3.cmn600_cache.upstream[24]	PVBusSlave
CMN_S3.cmn600_cache.upstream[25]	PVBusSlave
CMN_S3.cmn600_cache.upstream[26]	PVBusSlave
CMN_S3.cmn600_cache.upstream[27]	PVBusSlave
CMN_S3.cmn600_cache.upstream[28]	PVBusSlave
CMN_S3.cmn600_cache.upstream[29]	PVBusSlave
CMN_S3.cmn600_cache.upstream[2]	PVBusSlave
CMN_S3.cmn600_cache.upstream[30]	PVBusSlave
CMN_S3.cmn600_cache.upstream[31]	PVBusSlave
CMN_S3.cmn600_cache.upstream[32]	PVBusSlave
CMN_S3.cmn600_cache.upstream[33]	PVBusSlave
CMN_S3.cmn600_cache.upstream[34]	PVBusSlave
CMN_S3.cmn600_cache.upstream[35]	PVBusSlave
CMN_S3.cmn600_cache.upstream[36]	PVBusSlave
CMN_S3.cmn600_cache.upstream[37]	PVBusSlave
CMN_S3.cmn600_cache.upstream[38]	PVBusSlave
CMN_S3.cmn600_cache.upstream[39]	PVBusSlave

InstanceName	ComponentName
CMN_S3.cmn600_cache.upstream[3]	PVBusSlave
CMN_S3.cmn600_cache.upstream[40]	PVBusSlave
CMN_S3.cmn600_cache.upstream[41]	PVBusSlave
CMN_S3.cmn600_cache.upstream[42]	PVBusSlave
CMN_S3.cmn600_cache.upstream[43]	PVBusSlave
CMN_S3.cmn600_cache.upstream[44]	PVBusSlave
CMN_S3.cmn600_cache.upstream[45]	PVBusSlave
CMN_S3.cmn600_cache.upstream[46]	PVBusSlave
CMN_S3.cmn600_cache.upstream[47]	PVBusSlave
CMN_S3.cmn600_cache.upstream[48]	PVBusSlave
CMN_S3.cmn600_cache.upstream[49]	PVBusSlave
CMN_S3.cmn600_cache.upstream[4]	PVBusSlave
CMN_S3.cmn600_cache.upstream[50]	PVBusSlave
CMN_S3.cmn600_cache.upstream[51]	PVBusSlave
CMN_S3.cmn600_cache.upstream[52]	PVBusSlave
CMN_S3.cmn600_cache.upstream[53]	PVBusSlave
CMN_S3.cmn600_cache.upstream[54]	PVBusSlave
CMN_S3.cmn600_cache.upstream[55]	PVBusSlave
CMN_S3.cmn600_cache.upstream[56]	PVBusSlave
CMN_S3.cmn600_cache.upstream[57]	PVBusSlave
CMN_S3.cmn600_cache.upstream[58]	PVBusSlave
CMN_S3.cmn600_cache.upstream[59]	PVBusSlave
CMN_S3.cmn600_cache.upstream[5]	PVBusSlave
CMN_S3.cmn600_cache.upstream[60]	PVBusSlave
CMN_S3.cmn600_cache.upstream[61]	PVBusSlave
CMN_S3.cmn600_cache.upstream[62]	PVBusSlave
CMN_S3.cmn600_cache.upstream[63]	PVBusSlave
CMN_S3.cmn600_cache.upstream[64]	PVBusSlave
CMN_S3.cmn600_cache.upstream[65]	PVBusSlave
CMN_S3.cmn600_cache.upstream[66]	PVBusSlave
CMN_S3.cmn600_cache.upstream[67]	PVBusSlave
CMN_S3.cmn600_cache.upstream[68]	PVBusSlave
CMN_S3.cmn600_cache.upstream[69]	PVBusSlave
CMN_S3.cmn600_cache.upstream[6]	PVBusSlave
CMN_S3.cmn600_cache.upstream[70]	PVBusSlave
CMN_S3.cmn600_cache.upstream[71]	PVBusSlave
CMN_S3.cmn600_cache.upstream[72]	PVBusSlave
CMN_S3.cmn600_cache.upstream[73]	PVBusSlave
CMN_S3.cmn600_cache.upstream[74]	PVBusSlave

InstanceName	ComponentName
CMN_S3.cmn600_cache.upstream[75]	PVBusSlave
CMN_S3.cmn600_cache.upstream[76]	PVBusSlave
CMN_S3.cmn600_cache.upstream[77]	PVBusSlave
CMN_S3.cmn600_cache.upstream[78]	PVBusSlave
CMN_S3.cmn600_cache.upstream[79]	PVBusSlave
CMN_S3.cmn600_cache.upstream[7]	PVBusSlave
CMN_S3.cmn600_cache.upstream[80]	PVBusSlave
CMN_S3.cmn600_cache.upstream[81]	PVBusSlave
CMN_S3.cmn600_cache.upstream[82]	PVBusSlave
CMN_S3.cmn600_cache.upstream[83]	PVBusSlave
CMN_S3.cmn600_cache.upstream[84]	PVBusSlave
CMN_S3.cmn600_cache.upstream[85]	PVBusSlave
CMN_S3.cmn600_cache.upstream[86]	PVBusSlave
CMN_S3.cmn600_cache.upstream[87]	PVBusSlave
CMN_S3.cmn600_cache.upstream[88]	PVBusSlave
CMN_S3.cmn600_cache.upstream[89]	PVBusSlave
CMN_S3.cmn600_cache.upstream[8]	PVBusSlave
CMN_S3.cmn600_cache.upstream[90]	PVBusSlave
CMN_S3.cmn600_cache.upstream[91]	PVBusSlave
CMN_S3.cmn600_cache.upstream[92]	PVBusSlave
CMN_S3.cmn600_cache.upstream[93]	PVBusSlave
CMN_S3.cmn600_cache.upstream[94]	PVBusSlave
CMN_S3.cmn600_cache.upstream[95]	PVBusSlave
CMN_S3.cmn600_cache.upstream[96]	PVBusSlave
CMN_S3.cmn600_cache.upstream[97]	PVBusSlave
CMN_S3.cmn600_cache.upstream[98]	PVBusSlave
CMN_S3.cmn600_cache.upstream[99]	PVBusSlave
CMN_S3.cmn600_cache.upstream[9]	PVBusSlave
CMN_S3.cmn_s3_tag_cache	CMN_TAG_CACHE
CMN_S3.cmn_s3_tag_cache.metadata_controller0	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller0.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller1	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller1.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller10	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller10.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller100	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller100.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller101	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller101.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller102	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller102.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller103	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller103.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller104	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller104.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller105	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller105.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller106	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller106.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller107	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller107.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller108	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller108.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller109	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller109.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller11	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller11.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller110	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller110.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller111	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller111.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller112	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller112.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller113	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller113.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller114	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller114.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller115	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller115.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller116	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller116.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller117	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller117.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller118	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller118.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller119	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller119.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller12	MetaDataController



InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller12.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller120	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller120.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller121	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller121.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller122	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller122.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller123	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller123.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller124	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller124.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller125	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller125.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller126	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller126.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller127	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller127.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller13	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller13.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller14	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller14.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller15	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller15.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller16	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller16.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller17	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller17.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller18	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller18.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller19	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller19.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller2	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller2.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller20	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller20.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller21	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller21.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller22	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller22.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller23	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller23.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller24	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller24.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller25	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller25.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller26	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller26.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller27	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller27.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller28	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller28.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller29	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller29.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller3	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller3.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller30	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller30.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller31	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller31.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller32	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller32.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller33	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller33.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller34	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller34.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller35	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller35.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller36	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller36.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller37	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller37.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller38	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller38.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller39	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller39.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller4	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller4.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller40	MetaDataController

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller40.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller41	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller41.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller42	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller42.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller43	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller43.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller44	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller44.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller45	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller45.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller46	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller46.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller47	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller47.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller48	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller48.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller49	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller49.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller5	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller5.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller50	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller50.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller51	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller51.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller52	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller52.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller53	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller53.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller54	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller54.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller55	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller55.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller56	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller56.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller57	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller57.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller58	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller58.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller59	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller59.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller6	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller6.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller60	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller60.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller61	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller61.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller62	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller62.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller63	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller63.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller64	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller64.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller65	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller65.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller66	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller66.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller67	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller67.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller68	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller68.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller69	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller69.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller7	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller7.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller70	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller70.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller71	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller71.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller72	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller72.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller73	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller73.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller74	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller74.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller75	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller75.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller76	MetaDataController

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller76.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller77	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller77.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller78	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller78.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller79	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller79.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller8	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller8.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller80	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller80.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller81	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller81.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller82	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller82.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller83	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller83.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller84	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller84.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller85	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller85.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller86	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller86.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller87	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller87.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller88	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller88.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller89	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller89.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller9	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller9.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller90	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller90.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller91	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller91.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller92	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller92.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller93	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller93.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller94	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller94.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller95	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller95.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller96	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller96.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller97	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller97.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller98	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller98.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller99	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller99.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper0	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper1	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper10	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper100	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper101	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper102	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper103	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper104	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper105	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper106	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper107	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper108	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper109	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper11	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper110	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper111	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper112	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper113	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper114	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper115	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper116	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper117	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper118	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper119	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper12	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper120	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper121	PVBusMapper

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.remapper122	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper123	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper124	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper125	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper126	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper127	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper13	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper14	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper15	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper16	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper17	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper18	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper19	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper2	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper20	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper21	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper22	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper23	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper24	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper25	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper26	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper27	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper28	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper29	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper3	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper30	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper31	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper32	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper33	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper34	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper35	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper36	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper37	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper38	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper39	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper4	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper40	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper41	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper42	PVBusMapper

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.remapper43	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper44	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper45	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper46	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper47	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper48	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper49	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper5	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper50	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper51	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper52	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper53	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper54	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper55	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper56	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper57	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper58	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper59	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper6	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper60	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper61	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper62	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper63	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper64	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper65	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper66	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper67	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper68	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper69	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper7	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper70	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper71	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper72	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper73	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper74	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper75	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper76	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper77	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper78	PVBusMapper



InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.remapper79	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper8	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper80	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper81	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper82	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper83	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper84	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper85	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper86	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper87	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper88	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper89	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper9	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper90	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper91	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper92	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper93	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper94	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper95	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper96	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper97	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper98	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper99	PVBusMapper
CMN_S3.dmc_axu_mapper	PVBusMapper
CMN_S3.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN_S3.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor1	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor1.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor10	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor10.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor11	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor11.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor12	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor12.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor13	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor13.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor14	PVBusExclusiveMonitor

InstanceName	ComponentName
CMN_S3.hni_exclusive_monitor14.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor15	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor15.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor16	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor16.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor17	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor17.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor18	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor18.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor19	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor19.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor2	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor2.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor20	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor20.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor21	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor21.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor22	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor22.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor23	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor23.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor24	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor24.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor25	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor25.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor26	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor26.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor3	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor3.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor4	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor4.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor5	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor5.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor6	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor6.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor7	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor7.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor8	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor8.bus_mapper	PVBusMapper

InstanceName	ComponentName
CMN_S3.hni_exclusive_monitor9	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor9.bus_mapper	PVBusMapper
CMN_S3.ocm_decoder	PVBusMapper
CMN_S3.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN_S3.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN_S3.snf_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-919: CMN\_S3 MTI instances**

InstanceName	ComponentName
CMN_S3	CMN_S3
CMN_S3.bus_slave_ocm_NS	PVBusSlave
CMN_S3.bus_slave_ocm_S	PVBusSlave
CMN_S3.cmn600_cache	CMN600Cache
CMN_S3.cmn600_cache.upstream[0]	PVBusSlave
CMN_S3.cmn600_cache.upstream[100]	PVBusSlave
CMN_S3.cmn600_cache.upstream[101]	PVBusSlave
CMN_S3.cmn600_cache.upstream[102]	PVBusSlave
CMN_S3.cmn600_cache.upstream[103]	PVBusSlave
CMN_S3.cmn600_cache.upstream[104]	PVBusSlave
CMN_S3.cmn600_cache.upstream[105]	PVBusSlave
CMN_S3.cmn600_cache.upstream[106]	PVBusSlave
CMN_S3.cmn600_cache.upstream[107]	PVBusSlave
CMN_S3.cmn600_cache.upstream[108]	PVBusSlave
CMN_S3.cmn600_cache.upstream[109]	PVBusSlave
CMN_S3.cmn600_cache.upstream[10]	PVBusSlave
CMN_S3.cmn600_cache.upstream[110]	PVBusSlave
CMN_S3.cmn600_cache.upstream[111]	PVBusSlave
CMN_S3.cmn600_cache.upstream[112]	PVBusSlave
CMN_S3.cmn600_cache.upstream[113]	PVBusSlave
CMN_S3.cmn600_cache.upstream[114]	PVBusSlave
CMN_S3.cmn600_cache.upstream[115]	PVBusSlave
CMN_S3.cmn600_cache.upstream[116]	PVBusSlave
CMN_S3.cmn600_cache.upstream[11]	PVBusSlave
CMN_S3.cmn600_cache.upstream[12]	PVBusSlave
CMN_S3.cmn600_cache.upstream[13]	PVBusSlave
CMN_S3.cmn600_cache.upstream[14]	PVBusSlave
CMN_S3.cmn600_cache.upstream[15]	PVBusSlave
CMN_S3.cmn600_cache.upstream[16]	PVBusSlave

InstanceName	ComponentName
CMN_S3.cmn600_cache.upstream[17]	PVBusSlave
CMN_S3.cmn600_cache.upstream[18]	PVBusSlave
CMN_S3.cmn600_cache.upstream[19]	PVBusSlave
CMN_S3.cmn600_cache.upstream[1]	PVBusSlave
CMN_S3.cmn600_cache.upstream[20]	PVBusSlave
CMN_S3.cmn600_cache.upstream[21]	PVBusSlave
CMN_S3.cmn600_cache.upstream[22]	PVBusSlave
CMN_S3.cmn600_cache.upstream[23]	PVBusSlave
CMN_S3.cmn600_cache.upstream[24]	PVBusSlave
CMN_S3.cmn600_cache.upstream[25]	PVBusSlave
CMN_S3.cmn600_cache.upstream[26]	PVBusSlave
CMN_S3.cmn600_cache.upstream[27]	PVBusSlave
CMN_S3.cmn600_cache.upstream[28]	PVBusSlave
CMN_S3.cmn600_cache.upstream[29]	PVBusSlave
CMN_S3.cmn600_cache.upstream[2]	PVBusSlave
CMN_S3.cmn600_cache.upstream[30]	PVBusSlave
CMN_S3.cmn600_cache.upstream[31]	PVBusSlave
CMN_S3.cmn600_cache.upstream[32]	PVBusSlave
CMN_S3.cmn600_cache.upstream[33]	PVBusSlave
CMN_S3.cmn600_cache.upstream[34]	PVBusSlave
CMN_S3.cmn600_cache.upstream[35]	PVBusSlave
CMN_S3.cmn600_cache.upstream[36]	PVBusSlave
CMN_S3.cmn600_cache.upstream[37]	PVBusSlave
CMN_S3.cmn600_cache.upstream[38]	PVBusSlave
CMN_S3.cmn600_cache.upstream[39]	PVBusSlave
CMN_S3.cmn600_cache.upstream[3]	PVBusSlave
CMN_S3.cmn600_cache.upstream[40]	PVBusSlave
CMN_S3.cmn600_cache.upstream[41]	PVBusSlave
CMN_S3.cmn600_cache.upstream[42]	PVBusSlave
CMN_S3.cmn600_cache.upstream[43]	PVBusSlave
CMN_S3.cmn600_cache.upstream[44]	PVBusSlave
CMN_S3.cmn600_cache.upstream[45]	PVBusSlave
CMN_S3.cmn600_cache.upstream[46]	PVBusSlave
CMN_S3.cmn600_cache.upstream[47]	PVBusSlave
CMN_S3.cmn600_cache.upstream[48]	PVBusSlave
CMN_S3.cmn600_cache.upstream[49]	PVBusSlave
CMN_S3.cmn600_cache.upstream[4]	PVBusSlave
CMN_S3.cmn600_cache.upstream[50]	PVBusSlave
CMN_S3.cmn600_cache.upstream[51]	PVBusSlave

InstanceName	ComponentName
CMN_S3.cmn600_cache.upstream[52]	PVBusSlave
CMN_S3.cmn600_cache.upstream[53]	PVBusSlave
CMN_S3.cmn600_cache.upstream[54]	PVBusSlave
CMN_S3.cmn600_cache.upstream[55]	PVBusSlave
CMN_S3.cmn600_cache.upstream[56]	PVBusSlave
CMN_S3.cmn600_cache.upstream[57]	PVBusSlave
CMN_S3.cmn600_cache.upstream[58]	PVBusSlave
CMN_S3.cmn600_cache.upstream[59]	PVBusSlave
CMN_S3.cmn600_cache.upstream[5]	PVBusSlave
CMN_S3.cmn600_cache.upstream[60]	PVBusSlave
CMN_S3.cmn600_cache.upstream[61]	PVBusSlave
CMN_S3.cmn600_cache.upstream[62]	PVBusSlave
CMN_S3.cmn600_cache.upstream[63]	PVBusSlave
CMN_S3.cmn600_cache.upstream[64]	PVBusSlave
CMN_S3.cmn600_cache.upstream[65]	PVBusSlave
CMN_S3.cmn600_cache.upstream[66]	PVBusSlave
CMN_S3.cmn600_cache.upstream[67]	PVBusSlave
CMN_S3.cmn600_cache.upstream[68]	PVBusSlave
CMN_S3.cmn600_cache.upstream[69]	PVBusSlave
CMN_S3.cmn600_cache.upstream[6]	PVBusSlave
CMN_S3.cmn600_cache.upstream[70]	PVBusSlave
CMN_S3.cmn600_cache.upstream[71]	PVBusSlave
CMN_S3.cmn600_cache.upstream[72]	PVBusSlave
CMN_S3.cmn600_cache.upstream[73]	PVBusSlave
CMN_S3.cmn600_cache.upstream[74]	PVBusSlave
CMN_S3.cmn600_cache.upstream[75]	PVBusSlave
CMN_S3.cmn600_cache.upstream[76]	PVBusSlave
CMN_S3.cmn600_cache.upstream[77]	PVBusSlave
CMN_S3.cmn600_cache.upstream[78]	PVBusSlave
CMN_S3.cmn600_cache.upstream[79]	PVBusSlave
CMN_S3.cmn600_cache.upstream[7]	PVBusSlave
CMN_S3.cmn600_cache.upstream[80]	PVBusSlave
CMN_S3.cmn600_cache.upstream[81]	PVBusSlave
CMN_S3.cmn600_cache.upstream[82]	PVBusSlave
CMN_S3.cmn600_cache.upstream[83]	PVBusSlave
CMN_S3.cmn600_cache.upstream[84]	PVBusSlave
CMN_S3.cmn600_cache.upstream[85]	PVBusSlave
CMN_S3.cmn600_cache.upstream[86]	PVBusSlave
CMN_S3.cmn600_cache.upstream[87]	PVBusSlave

InstanceName	ComponentName
CMN_S3.cmn600_cache.upstream[88]	PVBusSlave
CMN_S3.cmn600_cache.upstream[89]	PVBusSlave
CMN_S3.cmn600_cache.upstream[8]	PVBusSlave
CMN_S3.cmn600_cache.upstream[90]	PVBusSlave
CMN_S3.cmn600_cache.upstream[91]	PVBusSlave
CMN_S3.cmn600_cache.upstream[92]	PVBusSlave
CMN_S3.cmn600_cache.upstream[93]	PVBusSlave
CMN_S3.cmn600_cache.upstream[94]	PVBusSlave
CMN_S3.cmn600_cache.upstream[95]	PVBusSlave
CMN_S3.cmn600_cache.upstream[96]	PVBusSlave
CMN_S3.cmn600_cache.upstream[97]	PVBusSlave
CMN_S3.cmn600_cache.upstream[98]	PVBusSlave
CMN_S3.cmn600_cache.upstream[99]	PVBusSlave
CMN_S3.cmn600_cache.upstream[9]	PVBusSlave
CMN_S3.cmn_s3_tag_cache	CMNTAGCACHECADI
CMN_S3.cmn_s3_tag_cache.metadata_controller0	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller0.MetadataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller1	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller1.MetadataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller10	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller10.MetadataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller100	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller100.MetadataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller101	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller101.MetadataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller102	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller102.MetadataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller103	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller103.MetadataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller104	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller104.MetadataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller105	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller105.MetadataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller106	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller106.MetadataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller107	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller107.MetadataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller108	MetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controller108.MetadataMapper	PVBusMapper

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller109	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller109.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller11	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller11.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller110	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller110.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller111	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller111.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller112	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller112.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller113	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller113.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller114	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller114.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller115	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller115.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller116	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller116.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller117	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller117.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller118	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller118.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller119	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller119.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller12	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller12.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller120	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller120.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller121	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller121.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller122	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller122.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller123	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller123.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller124	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller124.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller125	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller125.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller126	MetaDataController



InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller126.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller127	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller127.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller13	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller13.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller14	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller14.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller15	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller15.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller16	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller16.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller17	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller17.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller18	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller18.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller19	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller19.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller2	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller2.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller20	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller20.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller21	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller21.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller22	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller22.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller23	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller23.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller24	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller24.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller25	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller25.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller26	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller26.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller27	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller27.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller28	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller28.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller29	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller29.MetaDataMapper	PVBusMapper



InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller3	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller3.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller30	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller30.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller31	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller31.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller32	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller32.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller33	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller33.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller34	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller34.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller35	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller35.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller36	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller36.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller37	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller37.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller38	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller38.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller39	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller39.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller4	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller4.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller40	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller40.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller41	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller41.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller42	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller42.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller43	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller43.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller44	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller44.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller45	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller45.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller46	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller46.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller47	MetaDataController

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller47.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller48	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller48.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller49	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller49.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller5	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller5.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller50	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller50.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller51	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller51.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller52	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller52.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller53	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller53.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller54	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller54.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller55	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller55.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller56	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller56.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller57	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller57.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller58	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller58.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller59	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller59.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller6	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller6.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller60	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller60.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller61	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller61.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller62	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller62.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller63	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller63.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller64	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller64.MetaDataMapper	PVBusMapper

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller65	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller65.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller66	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller66.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller67	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller67.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller68	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller68.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller69	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller69.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller7	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller7.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller70	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller70.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller71	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller71.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller72	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller72.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller73	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller73.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller74	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller74.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller75	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller75.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller76	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller76.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller77	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller77.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller78	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller78.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller79	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller79.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller8	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller8.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller80	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller80.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller81	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller81.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller82	MetaDataController

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.metadata_controller82.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller83	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller83.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller84	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller84.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller85	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller85.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller86	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller86.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller87	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller87.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller88	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller88.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller89	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller89.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller9	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller9.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller90	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller90.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller91	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller91.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller92	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller92.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller93	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller93.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller94	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller94.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller95	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller95.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller96	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller96.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller97	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller97.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller98	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller98.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controller99	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controller99.MetaDataMapper	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper0	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper1	PVBusMapper

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.remapper10	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper100	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper101	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper102	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper103	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper104	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper105	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper106	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper107	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper108	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper109	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper11	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper110	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper111	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper112	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper113	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper114	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper115	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper116	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper117	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper118	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper119	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper12	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper120	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper121	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper122	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper123	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper124	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper125	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper126	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper127	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper13	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper14	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper15	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper16	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper17	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper18	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper19	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper2	PVBusMapper

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.remapper20	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper21	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper22	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper23	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper24	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper25	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper26	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper27	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper28	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper29	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper3	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper30	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper31	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper32	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper33	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper34	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper35	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper36	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper37	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper38	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper39	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper4	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper40	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper41	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper42	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper43	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper44	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper45	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper46	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper47	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper48	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper49	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper5	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper50	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper51	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper52	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper53	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper54	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper55	PVBusMapper

InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.remapper56	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper57	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper58	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper59	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper6	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper60	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper61	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper62	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper63	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper64	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper65	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper66	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper67	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper68	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper69	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper7	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper70	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper71	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper72	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper73	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper74	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper75	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper76	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper77	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper78	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper79	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper8	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper80	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper81	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper82	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper83	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper84	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper85	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper86	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper87	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper88	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper89	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper9	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper90	PVBusMapper



InstanceName	ComponentName
CMN_S3.cmn_s3_tag_cache.remapper91	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper92	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper93	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper94	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper95	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper96	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper97	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper98	PVBusMapper
CMN_S3.cmn_s3_tag_cache.remapper99	PVBusMapper
CMN_S3.dmc_axu_mapper	PVBusMapper
CMN_S3.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN_S3.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor1	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor1.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor10	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor10.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor11	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor11.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor12	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor12.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor13	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor13.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor14	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor14.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor15	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor15.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor16	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor16.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor17	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor17.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor18	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor18.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor19	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor19.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor2	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor2.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor20	PVBusExclusiveMonitor



InstanceName	ComponentName
CMN_S3.hni_exclusive_monitor20.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor21	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor21.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor22	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor22.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor23	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor23.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor24	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor24.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor25	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor25.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor26	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor26.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor3	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor3.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor4	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor4.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor5	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor5.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor6	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor6.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor7	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor7.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor8	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor8.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitor9	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitor9.bus_mapper	PVBusMapper
CMN_S3.ocm_decoder	PVBusMapper
CMN_S3.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN_S3.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN_S3.snf_mapper	PVBusMapper

## Ports for CMN\_S3

**Table 3-920: Ports**

Name	Protocol	Type	Description
event_clusters	Signal	Peer	CPU event communication signal from the clusters.
event_downstream_link_signal[44]	Signal	Master	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.

Name	Protocol	Type	Description
event_upstream_link_signal[44]	Signal	Slave	Event from the Hub towards the CMN
ic_dr_a4s	PVBus	Slave	Interrupt Controller Distributor-to-Remote AXI4Stream port.
ic_rd_a4s	PVBus	Master	Interrupt Controller Remote-to-Distributor AXI4Stream port.
intreqerrns_irq_out	Signal	Master	Interrupt signal
intreqerrs_irq_out	Signal	Master	Interrupt signal
intreqfaultns_irq_out	Signal	Master	Interrupt signal
intreqfaults_irq_out	Signal	Master	Interrupt signal
pvbus_m_cxs[44]	PVBus	Master	CXS downstream ports
pvbus_m_dmc_axu[16]	PVBus	Master	DMC AXU ports
pvbus_m_dsu_axu[256]	PVBus	Master	DSU AXU ports
pvbus_m_hni[256]	PVBus	Master	HNI downstream ports.
pvbus_m_mxp_axu[144]	PVBus	Master	MPX AXU ports
pvbus_m_snf[179]	PVBus	Master	SNF downstream port.
pvbus_s_apb	PVBus	Slave	APB interface port.
pvbus_s_ccg_apb[44]	PVBus	Slave	CCG APB interface port.
pvbus_s_cxs[44]	PVBus	Slave	CXS upstream ports
pvbus_s_rnf[296]	PVBus	Slave	RNF upstream ports.
pvbus_s_rni[270]	PVBus	Slave	RNI upstream ports.
reset_in	Signal	Slave	Reset signal.
rnf_sci_s[296]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in[296]	Signal	Slave	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s[120]	SystemCoherencyInterface	Slave	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in[270]	Signal	Slave	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rx_cxs_a4s[44]	PVBus	Slave	Receive channel of A4S packets from a remote CMN.
tx_cxs_a4s[44]	PVBus	Master	Transmit channel of A4S packets to a remote CMN.

## Parameters for CMN\_S3

### a4s\_logicalid

#### Type

string

#### Default value

0

A4S ID mapping of the GIC destination component connected through a CCG port.

Specify the `ccg_node_id` and the destination A4S Logical ID of the GIC component connected by using a decimal number format like:

```
<CCG_NODEID0>=<A4S_LID0>,<CCG_NODEID1>=<A4S_LID1>
```

For example for CCG Node ID 54 with A4S ID 12 - 54=12.

All of the CCG nodes must be specified.

The parameter is only valid when the `enable_a4s` is also enabled. The default behavior without this parameter is to automatically assign an incrementing A4S ID

### **acchannelen\_rnf**

#### **Type**

string

#### **Default value**

"0"



Note

DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each `rnf` port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`

### **acchannelen\_rni**

#### **Type**

string

#### **Default value**

"0"



Note

DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`

**cache\_state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Model the cache state.

**cmn\_s3\_tag\_cache.metadata\_controller.init\_value****Type**

int

**Default value**

0xd

**Description**

Initialize metadata memory with this value. If one of init\_values\_json or init\_values\_json\_file is specified, this value applies only to any metadata not specified in the JSON.

**cmn\_s3\_tag\_cache.metadata\_controller.init\_values\_json****Type**

string

**Default value**

""

**Description**

A JSON value describing initial metadata values. Mutually exclusive with init\_values\_json\_file. The format is as follows: {"regions": [{ "begin": 0x0, "end": 0x10000, "mte\_tag": 0xa }, {"begin": 0x20000, "end": 0x50000, "mte\_tag": 0xc }]}.

**cmn\_s3\_tag\_cache.metadata\_controller.init\_values\_json\_file****Type**

string

**Default value**

""

**Description**

Path to a JSON file with initial metadata values. Mutually exclusive with init\_values\_json. The format is as follows: {"regions": [{ "begin": 0x0, "end": 0x10000, "mte\_tag": 0xa }, {"begin": 0x20000, "end": 0x50000, "mte\_tag": 0xc }]}.

**cmn\_s3\_tag\_cache.metadata\_controller.is\_enabled****Type**

bool

**Default value**

0x0

**Description**

If false, disables the MetaData controller functionality, and makes the component invisible to passing transactions.

**cmn\_s3\_tag\_cache.metadata\_controller.mte\_tag\_carveout\_json****Type**

string

**Default value**

""

**Description**

JSON string that specifies the PA range of the tag carveout regions and the beginning of the PA range for which they provide tag storage. If `pa_regions_with_metadata_storage` defines which regions can have metadata, the tag carveout regions cannot overlap them, and each tagged region must be entirely covered by one of them. The block size must be  $\geq 64$  bytes and a power of 2, defaulting to 4KiB. The maximum block size supported is 4KiB. The carveout region size must be  $\geq 4$ KiB and a power of 2, and determines the size of the corresponding tagged region. { "regions": [ { "begin": 0x0, "tag\_carveout\_region": [0xfffff00000, 0xfffff00fff]}, { "begin": 0x20000, "tag\_carveout\_region": [0xfffff01000, 0xfffff01fff], "block\_size": 0x100}, { "begin": 0x100000, "tag\_carveout\_region": [0xfffff08000, 0xfffff0Bfff], "block\_size": 0x2000}]}.

**cmn\_s3\_tag\_cache.metadata\_controller.mte\_tag\_carveout\_json\_file****Type**

string

**Default value**

""

**Description**

Path to a JSON file that specifies the PA range of the tag carveout regions with the same format as `mte_tag_carveout_json`. Only one of `mte_tag_carveout_json` and `mte_tag_carveout_json_file` can be used. .

**cmn\_s3\_tag\_cache.metadata\_controller.mte\_tag\_carveout\_tag\_order****Type**

string

**Default value**

little-endian

**Description**

Order of the tags within the MTE tag carveout blocks. This can be little-endian (same order as the corresponding tagged data) or big-endian (reverse order). The parameter accepts both '-' and '\_', so 'little-endian', 'big-endian', 'little\_endian' and 'big\_endian' are all valid. .

**cmn\_s3\_tag\_cache.metadata\_controller.pa\_regions\_with\_metadata\_storage****Type**

string

**Default value**

""

**Description**

Specify the address region where the metadata storage is available for each PAS in a JSON format. If the PAS does not have a region specified, the PAS has metadata storage for all of the space. The regions are defined by begin and end\_incl addresses. Example: { "ns": [0xa0000000, 0xa0000fff], "s": [0xb0000000, 0xb0000fff], "rl": [0xc0000000, 0xc0000fff], "rt": [0xd0000000, 0xd0000fff] } ns: non-secure, s: secure, rl: realm, rt: root.

**debug\_force\_snoop****Type**

bool

**Default value**

false

The CMN\_S3 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.



This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

**dmc\_periphbase****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Value for DMC\_PERIPHBASE.

**dsu\_periphbase****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Value for DSU\_PERIPHBASE.

**enable\_a4s****Type**

bool

**Default value**

0x0

**Description**

Enables A4S ports for GIC multi-chip routing.

**enable\_logger****Type**

bool

**Default value**

0x0

**Description**

Enable PVBUSLoggers for the downstream ports in the CMN model.

**enable\_ras****Type**

bool

**Default value**

0x0

**Description**

Enables RAS. There is an impact on performance when RAS is enabled.

**enable\_rnsam\_to\_hnf\_wider\_hash****Type**

bool

**Default value**

false

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

### **`force_rnsam_internal`**

#### **Type**

`bool`

#### **Default value**

`0x1`

#### **Description**

Force all RNSAMs to be internal independently of the mesh topology.

### **`hnf_mpam_idr_override`**

#### **Type**

`uint64_t`

#### **Default value**

`0`

Set to override `hnf_mpam_idr[31:24]` value. Bit[28] is Reserved and is ignored.

### **`legacy_tz_en`**

#### **Type**

`bool`

#### **Default value**

`false`

When set: Root registers accessible from Secure.

Realm Registers accessible from Non-Secure.

RCR are **RAZ/WI**.

### **`mesh_config_file`**

#### **Type**

`string`

#### **Default value**

`""`

#### **Description**

Name of a file containing mesh placement of CMN\_S3 components.



**mxp\_axu\_periphbase****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Value for MXP\_AXU\_PERIPHBASE.

**partner\_param0****Type**

int

**Default value**

0x0

**Description**

Partner Param.

**periphbase****Type**

uint64\_t

**Default value**

0x20000000

Value for PERIPHBASE. Bits [27:0] are treated 0

**print\_cmn\_ccix\_config****Type**

bool

**Default value**

0x0

**Description**

Print information about the CCIX configuration.

**print\_cmn\_config****Type**

bool

**Default value**

0x0

**Description**

Print the mesh topology and children pointers acquired from the YAML file.

**register\_traces\_for\_ccg\_apb\_accesses**

**Type**  
bool

**Default value**  
false



Will be removed when enhancement SDDKW-74284 is done.

---

Intended for use with trace plugins.

**true**  
registers traces to CCG register accesses through CCG APB interface.

**false**  
registers traces to CMN register accesses through all other interfaces (eg RN nodes).

**revision**

**Type**  
string

**Default value**  
"r0p0"

Component revision.

Currently supports r0p0, r0p1, r2p0.

**rnf\_sci\_enable**

**Type**  
string

**Default value**  
"0x0"

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

- 1**  
Managed by SCI
- 0**  
Managed by Software

The input value is a string, for example 0xffff or ffff

**rni\_sci\_enable****Type**

string

**Default value**

"0x0"

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example 0xffff or ffff

**show\_banner****Type**

uint64\_t

**Default value**

2

Show component banner:

**0**

supress entire banner

**1**

suppress config file

**2+**

show full banner

**skip\_cmn\_config\_check****Type**

bool

**Default value**

0x0

**Description**

Skip any topology configuration checks. The maximum number of devices per type not verified.

**use\_yaml\_periphbase****Type**

bool

**Default value**

false

Use yml param `CFGM_PERIPHBASE_PARAM` to specify periphbase address.

If false, model parameter `periphbase` will be used.

**yml\_has\_node\_addresses****Type**

bool

**Default value**

0x0

**Description**

Does the top-level YML file describe node-addresses ?.

### 3.10.21 DCSU

Diagnostic Control and Status Unit. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-921: IP revisions support**

Revision	Quality level
0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Iris and MTI instances for DCSU**

This model has the following Iris instances:

**Table 3-922: DCSU Iris instances**

InstanceName	ComponentName
DCSU	DCSU

**Ports for DCSU****Table 3-923: Ports**

Name	Protocol	Type	Description
apb_pvbus_s	PVBus	Slave	-
dcsu_irq_out	Signal	Master	-
diag_dma_gpo_ch_in[16]	Value	Slave	-
diag_lcs_in	Value	Slave	-
diag_lcs_valid_in	Signal	Slave	-

Name	Protocol	Type	Description
diag_psi_dcu_en0_in[96]	Signal	Slave	-
diag_psi_dcu_en1_in[6]	Signal	Slave	-
diag_psi_gppc_in[16]	Signal	Slave	-
diag_psi_psi_status_in[32]	Signal	Slave	-
diag_psi_sam_in[64]	Signal	Slave	-
die_id_out	Value	Master	-
poreset	Signal	Slave	-
post_code_out	Value	Master	-

## Parameters for DCSU

### diagnostics

#### Type

int

#### Default value

0x2

#### Description

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

### die\_id

#### Type

int

#### Default value

0x0

#### Description

DIE ID register and port value.

## 3.10.22 DMA350

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-924: IP revisions support**

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About DMA350

This model supports the following functionality:

- 1-8 DMA channels
- 1D memory copy including increments, auto-reload, and command linking
- Interrupt capability for each channel
- 2D memory copy
- 1DWRAP and 2DWRAP support
- Template-based pack and unpack capability
- Security settings per channel
- Trigger input and output ports selectable for each channel
- General Purpose Output (GPO) per channel
- Streaming input and output interfaces per channel
- ADDR\_WIDTH and DATA\_WIDTH can be 32 bits or 64 bits

## Iris and MTI instances for DMA350

This model has the following Iris instances:

**Table 3-925: DMA350 Iris instances**

InstanceName	ComponentName
DMA350	DMA350
DMA350.pvbus_m0_bus_master	PVBusMaster
DMA350.pvbus_m1_bus_master	PVBusMaster
DMA350.pvbus_s_bus_slave	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_0	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_1	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_10	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_11	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_12	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_13	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_14	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_15	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_2	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_3	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_4	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_5	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_6	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_7	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_8	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_9	PVBusSlave

InstanceName	ComponentName
DMA350.pvbus_stream_out_bus_master_ch_0	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_1	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_10	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_11	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_12	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_13	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_14	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_15	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_2	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_3	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_4	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_5	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_6	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_7	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_8	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_9	PVBusMaster

This model has the following MTI trace components:

**Table 3-926: DMA350 MTI instances**

InstanceName	ComponentName
DMA350	DMA350
DMA350.pvbus_m0_bus_master	PVBusMaster
DMA350.pvbus_m1_bus_master	PVBusMaster
DMA350.pvbus_s_bus_slave	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_0	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_1	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_10	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_11	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_12	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_13	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_14	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_15	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_2	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_3	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_4	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_5	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_6	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_7	PVBusSlave
DMA350.pvbus_stream_in_bus_slave_ch_8	PVBusSlave

InstanceName	ComponentName
DMA350.pvbus_stream_in_bus_slave_ch_9	PVBusSlave
DMA350.pvbus_stream_out_bus_master_ch_0	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_1	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_10	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_11	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_12	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_13	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_14	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_15	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_2	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_3	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_4	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_5	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_6	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_7	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_8	PVBusMaster
DMA350.pvbus_stream_out_bus_master_ch_9	PVBusMaster

## Ports for DMA350

**Table 3-927: Ports**

Name	Protocol	Type	Description
allch_pause_ack_nonsec	Signal	Master	-
allch_pause_ack_sec	Signal	Master	-
allch_pause_req_nonsec	Signal	Slave	Channel pause req/ack for all nonsecure channels
allch_pause_req_sec	Signal	Slave	Channel pause req/ack for all secure channels
allch_stop_ack_nonsec	Signal	Master	-
allch_stop_ack_sec	Signal	Master	-
allch_stop_req_nonsec	Signal	Slave	Channel stop req/ack for all nonsecure channels
allch_stop_req_sec	Signal	Slave	Channel stop req/ack for all secure channels
boot_addr	Value_64	Slave	Address when boot_en is enabled
boot_en	Signal	Slave	Enables channel 0 to load first command after reset from boot_addr
boot_memattr	Value	Slave	Memory attribute setting for the boot_addr
boot_shareattr	Value	Slave	Shareability attribute for the boot_attr
ch_enabled[16]	Signal	Master	Enable status indicator per channel
ch_err[16]	Signal	Master	Error status indicator per channel
ch_nonsec[16]	Signal	Master	Nonsecure status indicator per channel
ch_paused[16]	Signal	Master	Paused status indicator per channel
ch_priv[16]	Signal	Master	Privilege status indicator per channel



Name	Protocol	Type	Description
ch_stopped[16]	Signal	Master	Stopped status indicator per channel
clk_in	ClockSignal	Slave	Ada DMA clock
gpo_ch[16]	Value	Master	MISC signals General purpose output for channels 0-15 Index refers to the channel
irq_channel[16]	Signal	Master	Channel IRQ Signals
irq_comb_nonsec	Signal	Master	Nonsecure IRQ Signal
irq_comb_nonsec_err	Signal	Master	Nonsecure error IRQ Signal
irq_comb_sec	Signal	Master	Secure IRQ Signal
irq_comb_sec_err	Signal	Master	Secure error IRQ Signal
irq_sec_viol_err	Signal	Master	Security violation IRQ Signal
pvbus_m0	PVBus	Master	AXI5 Master 0 Interface
pvbus_m1	PVBus	Master	AXI5 Master 1 Interface
pvbus_s	PVBus	Slave	APB4 Slave Interface
pvbus_stream_in[16]	PVBus	Slave	AXI-Stream In Interface
pvbus_stream_out[16]	PVBus	Master	AXI-Stream Out Interface
reset_in	Signal	Slave	Ada DMA asynchronous reset
trig_in_ack[32]	Signal	Master	Trigger In Acknowledgement Interface
trig_in_ack_type[32]	Value	Master	-
trig_in_req[32]	Signal	Slave	Trigger In Request Interface
trig_in_req_type[32]	Value	Slave	-
trig_out_ack[32]	Signal	Slave	Trigger Out Acknowledgement Interface
trig_out_req[32]	Signal	Master	Trigger Out Request Interface

## Parameters for DMA350

### ADDR\_WIDTH

#### Type

int

#### Default value

0x20

#### Description

Address width of the bus interface.

### AXI5\_M1\_ADDRESS\_RANGES

#### Type

string

#### Default value

""

**Description**

Address ranges for AXI5 M1 interface in the format e.g. [{"begin":0x40000000, "size":0x1000}, {"begin":0x80000000, "size":0x2000}]. Default when not specified uses AXI5 M0 interface.

**AXI5\_M1\_PRESENT****Type**

bool

**Default value**

0x0

**Description**

Enables an additional master port. When set the m1 master port is present on the top level port list and additional include file can be used with a System Verilog function that defines which address ranges are mapped to the m1 interface.

**CHID\_WIDTH****Type**

int

**Default value**

0x0

**Description**

Width of the configurable channel ID user signal. When set to 0, then the archid and awchid ports are not present on the module.

**CH\_0\_FIFO\_DEPTH****Type**

int

**Default value**

0x2

**Description**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1. This relates to the maximum burst size a channel can support which can be calculated as  $\min(16, (\text{FIFO\_DEPTH}+1)/2)$ . This setting needs to be aligned with the bandwidth requirements of the selected channel. This parameter highly affects the area of the design.

**CH\_1\_FIFO\_DEPTH****Type**

int

**Default value**

0x2

**Description**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1. This relates to the maximum burst size a channel can support which can be calculated as  $\min(16, (\text{FIFO\_DEPTH}+1)/2)$ . This setting needs to be aligned with the bandwidth requirements of the selected channel. This parameter highly affects the area of the design.

**CH\_2\_FIFO\_DEPTH****Type**

int

**Default value**

0x2

**Description**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1. This relates to the maximum burst size a channel can support which can be calculated as  $\min(16, (\text{FIFO\_DEPTH}+1)/2)$ . This setting needs to be aligned with the bandwidth requirements of the selected channel. This parameter highly affects the area of the design.

**CH\_3\_FIFO\_DEPTH****Type**

int

**Default value**

0x2

**Description**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1. This relates to the maximum burst size a channel can support which can be calculated as  $\min(16, (\text{FIFO\_DEPTH}+1)/2)$ . This setting needs to be aligned with the bandwidth requirements of the selected channel. This parameter highly affects the area of the design.

**CH\_4\_FIFO\_DEPTH****Type**

int

**Default value**

0x2

**Description**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1. This relates to the maximum burst size a channel can support which can be calculated as  $\min(16, (\text{FIFO\_DEPTH}+1)/2)$ . This setting needs to be aligned with the bandwidth requirements of the selected channel. This parameter highly affects the area of the design.

**CH\_5\_FIFO\_DEPTH****Type**

int

**Default value**

0x2

**Description**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1. This relates to the maximum burst size a channel can support which can be calculated as  $\min(16, (\text{FIFO\_DEPTH}+1)/2)$ . This setting needs to be aligned with the bandwidth requirements of the selected channel. This parameter highly affects the area of the design.

**CH\_6\_FIFO\_DEPTH****Type**

int

**Default value**

0x2

**Description**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1. This relates to the maximum burst size a channel can support which can be calculated as  $\min(16, (\text{FIFO\_DEPTH}+1)/2)$ . This setting needs to be aligned with the bandwidth requirements of the selected channel. This parameter highly affects the area of the design.

**CH\_7\_FIFO\_DEPTH****Type**

int

**Default value**

0x2

**Description**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1. This relates to the maximum burst size a channel can support which can be calculated as  $\min(16, (\text{FIFO\_DEPTH}+1)/2)$ . This setting needs to be aligned with the bandwidth requirements of the selected channel. This parameter highly affects the area of the design.

**CH\_EXT\_FEAT\_EN****Type**

bool

**Default value**

0x1

**Description**

Enabling the extended feature set for each channel. The extension contains 2D, WRAP, TMPLT features. Default value enables it for the number of channels.

**CH\_GPO\_EN****Type**

bool

**Default value**

0x1

**Description****CH\_GPO\_MASK****Type**

int

**Default value**

0x0

**Description**

A bitmask for enabling the GPO port for each channel. The width of the bitmask is NUM\_CHANNELS-1. When bit n is set to 1 then the GPO is enabled for channel n and the gpo\_ch\_n[GPO\_WIDTH-1:0] port appears on the module.

**CH\_STREAM\_EN****Type**

bool

**Default value**

0x1

**Description****CH\_STREAM\_MASK****Type**

int

**Default value**

0x0

**Description**

A bitmask for enabling the stream interfaces for each channel. The width of the bitmask is NUM\_CHANNELS-1. When bit n is set to 1 then the stream interfaces are enabled for channel n and the relevant ports appears on the module. NOTE: When streaming interface is enabled the actual FIFO size of the channel will be the double of CH\_<N>\_FIFO\_DEPTH.

**DATA\_WIDTH****Type**

int

**Default value**

0x40

**Description**

Data width of the bus interface.

**DISABLE\_DEVICE****Type**

bool

**Default value**

0x0

**Description**

Disable device and ignore all interfaces.

**DUMP\_CONFIG****Type**

bool

**Default value**

0x0

**Description**

Display DMA-350 DMAC parameters.

**GPO\_WIDTH****Type**

int

**Default value**

0x1

**Description**

Width of GPO output for every channel. When multiple channels have GPOs then the width must be set to the maximum number of GPOs a channel can have, and unused GPO ports need to be left unconnected. When all bits of CH\_GPO\_MASK is 0, this parameter is not relevant.

**NUM\_CHANNELS****Type**

int

**Default value**

0x2

**Description**  
Number of configurable DMA channels.

**NUM\_TRIGGER\_IN**

**Type**  
int

**Default value**  
0x2

**Description**  
Number of trigger input ports.

**NUM\_TRIGGER\_OUT**

**Type**  
int

**Default value**  
0x2

**Description**  
Number of trigger output ports.

**SECEXT\_PRESENT**

**Type**  
bool

**Default value**  
0x1

**Description**  
Enables Trustzone security support.

3.10.23 DMC500

ARM Dynamic Memory Controller(DMC500). This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-928: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About DMC500

A platform can have multiple instances of this component. For example:

```
//LISA instantiation
composition
{
    // Memory controllers
    dmc0      : DMC500("default_region_attributes"=dmc_default_region_attributes,
                      "default_region_id_access"=dmc_default_region_id_access,
                      "passthrough_debug_access"=true);
    dmc1      : DMC500("default_region_attributes"=dmc_default_region_attributes,
                      "default_region_id_access"=dmc_default_region_id_access,
                      "passthrough_debug_access"=true);
}
```

## Differences between the model and the RTL

The model has the following limitations:

- It does not support address striping.
- It works with linear addresses and not in rank,bank,row,column form.
- It does not include any mechanism for error injection or detection.
- Scrubbing functionality is only provided from the interface point of view.
- It does not implement direct read or write commands.
- It does not implement any performance counters.
- All OR'd interrupt signals are missing from this release of the model. Users can connect the failed access interrupt as a substitute.
- The model combines separate failed access interrupts for system interfaces 1 and 2 into a single failed access interrupt.
- DMC-500 has three separate reset signals whereas this model has a single reset signal which supports the combined assertion of three resets. This model does not support separate reset signals.

## Iris and MTI instances for DMC500

This model has the following Iris instances:

**Table 3-929: DMC500 Iris instances**

InstanceName	ComponentName
DMC500	DMC500
DMC500.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-930: DMC500 MTI instances**

InstanceName	ComponentName
DMC500	DMC-500
DMC500.busslave	PVBusSlave



## Ports for DMC500

**Table 3-931: Ports**

Name	Protocol	Type	Description
apb_pvbus_s	PVBus	Slave	Programmers interface to program and control the DMC-500.
failed_access_interrupt_signal	Signal	Master	The DMC has detected a system request that has failed a permissions check and a previously detected assertion was not cleared.
filter_pvbus_m	PVBus	Master	DMC master port from System Interface 0 to memory.
filter_pvbus_s	PVBus	Slave	System interface 0. Generally, Non-coherent Interface.
reset_signal	Signal	Slave	DMC reset.
si1_filter_pvbus_m	PVBus	Master	DMC master port from System Interface 1 to memory.
si1_filter_pvbus_s	PVBus	Slave	System interface 1. Generally, Coherent Interface.

## Parameters for DMC500

### **default\_region\_attributes**

#### Type

int

#### Default value

0x1

#### Description

Default Region Secure attributes. Only bits 31,30 set Secure RD/WR enable.

### **default\_region\_id\_access**

#### Type

int

#### Default value

0x0

#### Description

Default Region NSAID permissions. Bits 31-16 set non-secure WR enable and bits 15-0 set non-secure RD enable.

### **passthrough\_debug\_access**

#### Type

bool

#### Default value

0x0

#### Description

Always allow debug access to memory.

### 3.10.24 DMC520

ARM Dynamic Memory Controller(DMC520). This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-932: IP revisions support**

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About DMC520

A platform can have multiple instances of this component. For example:

```
//LISA instantiation
composition
{
    // Memory controllers
    dmc520_0      : DMC520("passthrough_debug_access"=true);
    dmc520_1      : DMC520("passthrough_debug_access"=true);
}
```

#### Limitations

- The model does not support address striping.
- It works with linear addresses and not in rank, bank, row, column form.
- It does not include any mechanism for error injection or detection.
- Scrubbing functionality is only provided from the interface point of view.
- It does not implement direct read or write commands.
- It does not implement any performance counters.

#### Differences between the model and the RTL

The DMC520 and DMC620 models have different interfaces to those in the hardware due to the level of abstraction of memory in Fast Models. These are the differences:

- Like the hardware, the model has a slave port for configuring register accesses, `apb_pvbuss`, and an AXI interface for incoming memory transactions that are attempting to access memory that is managed by the DMC.
- The hardware component translates incoming transactions on the AXI interface to a format that is conducive to accessing DRAM chips. The model performs TrustZone access control and models the DMC readiness state, but does not translate the transactions. If allowed, the model forwards incoming transactions to be handled by a memory storage handling component that works at the transaction level.

#### Iris and MTI instances for DMC520

This model has the following Iris instances:

**Table 3-933: DMC520 Iris instances**

InstanceName	ComponentName
DMC520	DMC520
DMC520.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-934: DMC520 MTI instances**

InstanceName	ComponentName
DMC520	DMC-520
DMC520.busslave	PVBusSlave

## Ports for DMC520

**Table 3-935: Ports**

Name	Protocol	Type	Description
all_or_interrupt_signal	Signal	Master	A combined interrupt that is the logical OR of the other interrupts.
apb_pvbus_s	PVBus	Slave	Programmers interface to program and control the DMC-520.
arch_fsm_interrupt_signal	Signal	Master	The DMC has detected a change in the architectural state.
failed_access_interrupt_signal	Signal	Master	The DMC has detected a system request that has failed a permissions check and a previously detected assertion was not cleared.
filter_pvbus_m	PVBus	Master	DMC master port to memory.
filter_pvbus_s	PVBus	Slave	System interface.
reset_signal	Signal	Slave	DMC reset.
scrub_event_in[8]	Signal	Slave	Scrub event n trigger.
scrub_event_out[8]	Signal	Master	Scrub event n triggered.

## Parameters for DMC520

### **override\_default\_config**

#### Type

bool

#### Default value

0x0

#### Description

Override default block-all behavior of DMC. Allow access to memory.

### **passthrough\_debug\_access**

#### Type

bool

**Default value**

0x0

**Description**

Always allow debug access to memory.

**3.10.25 DMC620**

ARM Dynamic Memory Controller(DMC620). This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-936: IP revisions support**

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Limitations**

- It does not support address striping.
- It works with linear addresses and not in rank,bank,row,column form.
- It includes error injection and detection mechanisms and syndrome registers support only for RAS error types 4 (ECC single-bit SRAM error) and 5 (ECC double-bit SRAM error).
- Scrubbing functionality is not provided.
- It does not implement direct read or write commands.
- It does not implement any performance counters.

**Differences between the model and the RTL**

The DMC520 and DMC620 models have different interfaces to those in the hardware due to the level of abstraction of memory in Fast Models. These are the differences:

- Like the hardware, the model has a slave port for configuring register accesses, `apb_pvbuss_s`, and an AXI interface for incoming memory transactions that are attempting to access memory that is managed by the DMC.
- The hardware component translates incoming transactions on the AXI interface to a format that is conducive to accessing DRAM chips. The model performs TrustZone access control and models the DMC readiness state, but does not translate the transactions. If allowed, the model forwards incoming transactions to be handled by a memory storage handling component that works at the transaction level.

**Iris and MTI instances for DMC620**

This model has the following Iris instances:

**Table 3-937: DMC620 Iris instances**

InstanceName	ComponentName
DMC620	DMC620
DMC620.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-938: DMC620 MTI instances**

InstanceName	ComponentName
DMC620	DMC-620
DMC620.busslave	PVBusSlave

## Ports for DMC620

**Table 3-939: Ports**

Name	Protocol	Type	Description
all_or_interrupt_signal	Signal	Master	A combined interrupt that is the logical OR of the other interrupts.
apb_pvbus_s	PVBus	Slave	Programmers interface to program and control the DMC-620.
arch_fsm_interrupt_signal	Signal	Master	The DMC has detected a change in the architectural state.
failed_access_interrupt_signal	Signal	Master	The DMC has detected a system request that has failed a permissions check and a previously detected assertion was not cleared.
filter_pvbus_m	PVBus	Master	DMC master port to memory.
filter_pvbus_s	PVBus	Slave	System interface.
interrupt_cfh_master	Signal	Master	The DMC has detected and corrected a single bit error on the RAM access.
interrupt_combined_oflow_master	Signal	Master	The DMC has detected a counter overflow.
interrupt_fh_master	Signal	Master	The DMC has detected a double bit error on the RAM access.
reset_signal	Signal	Slave	DMC reset.

## Parameters for DMC620

### **override\_default\_config**

#### Type

bool

#### Default value

0x0

#### Description

Override default block-all behavior of DMC. Allow access to memory.

**passthrough\_debug\_access**

**Type**

bool

**Default value**

0x0

**Description**

Always allow debug access to memory.

**3.10.26 DMC\_400**

ARM PrimeCell Dynamic Memory Controller(DMC400). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-940: IP revisions support**

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About DMC\_400**

The configuration of this model by setting the registers does not generally affect accesses to main memory.

This component has no timing information, so changing the values of the timing registers has no effect on behavior. The memory models do not attach to the component, and error checking does not update registers because the model does not include the possibility of errors.

**Iris and MTI instances for DMC\_400**

This model has the following Iris instances:

**Table 3-941: DMC\_400 Iris instances**

InstanceName	ComponentName
DMC_400	DMC_400
DMC_400.apb_slave	PVBusSlave
DMC_400.ex_mon0	PVBusMapper
DMC_400.ex_mon1	PVBusMapper
DMC_400.ex_mon2	PVBusMapper
DMC_400.ex_mon3	PVBusMapper

This model has the following MTI trace components:

Table 3-942: DMC\_400 MTI instances

InstanceName	ComponentName
DMC_400.apb_slave	PVBusSlave
DMC_400.ex_mon0	PVBusMapper
DMC_400.ex_mon1	PVBusMapper
DMC_400.ex_mon2	PVBusMapper
DMC_400.ex_mon3	PVBusMapper

Ports for DMC\_400

Table 3-943: Ports

Name	Protocol	Type	Description
apb_interface	PVBus	Slave	Slave bus interface for register access.
axi_if_in[4]	PVBus	Slave	Slave bus for connecting to bus decoder.
axi_if_out[4]	PVBus	Master	Master to connect to DRAM.
clr_ex_mon	Signal	Master	Indicates when global monitors state is cleared.
user_status_ext	Value	Slave	Allow user status to be set from outside.

Parameters for DMC\_400

ECC\_SUPPORT

Type

bool

Default value

0x1

Description

Does the controller support ECC?.

IF\_CHIP0

Type

int

Default value

0xffffffffffffffffffff

Description

Set this parameter to 0 if memory is connected.

IF\_CHIP1

Type

int

Default value

0xffffffffffffffffffff

**Description**

Set this parameter to 0 if memory is connected.

**IF\_CHIP2****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Set this parameter to 0 if memory is connected.

**IF\_CHIP3****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Set this parameter to 0 if memory is connected.

**MEMORY\_WIDTH****Type**

int

**Default value**

0x20

**Description**

Valid widths are 16, 32 or 64 bits.

**diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

**revision\_string****Type**

string

**Default value**

"rOp1"



**Description**

Revision.

### 3.10.27 EthosU55

Arm Ethos-U55 microNPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-944: IP revisions support**

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About EthosU55**

- The EthosU55 model does not expose its registers through CADI or Iris.
- The `resetn_in` signal is active-LOW.

**Iris and MTI instances for EthosU55**

This model has the following Iris instances:

**Table 3-945: EthosU55 Iris instances**

InstanceName	ComponentName
EthosU55	EthosU55
EthosU55.pvbusmaster0	PVBusMaster
EthosU55.pvbusmaster1	PVBusMaster
EthosU55.pvbusslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-946: EthosU55 MTI instances**

InstanceName	ComponentName
EthosU55.pvbusmaster0	<a href="#">PVBusMaster</a>
EthosU55.pvbusmaster1	<a href="#">PVBusMaster</a>
EthosU55.pvbusslave	<a href="#">PVBusSlave</a>

**Ports for EthosU55****Table 3-947: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	NPU clock signal
irq_out	<a href="#">Signal</a>	Master	Sends interrupt requests to the external host application processor

Name	Protocol	Type	Description
popl_in	Signal	Slave	The Power-On-Reset Privilege Level (PORPL). LOW means User level. HIGH means Privileged level
posl_in	Signal	Slave	The Power-On-Reset Security Level (PORSL). LOW means Secure. HIGH means Non-secure.
pvbuss_m0	PVBus	Master	Port 0 for NPU to access external memory
pvbuss_m1	PVBus	Master	Port 1 for NPU to access external memory
pvbuss_s	PVBus	Slave	Port to access NPU control registers
resetsn_in	Signal	Slave	NPU reset signal

## Parameters for EthosU55

### diagnostics

#### Type

int

#### Default value

0x0

#### Description

Enables extra information messages from the component about the NPU configuration and the PORPL/PORSL port values. Use any nonzero integer to enable all messages or 0 to disable them.

### extra\_args

#### Type

string

#### Default value

""

#### Description

To activate fast processing mode, which significantly improves the performance of the model, set this parameter to "--fast". In fast mode, NPU performance counters are not representative of counters on real hardware. We recommend you do not use any other value without instructions from Arm Technical Support.

### num\_macs

#### Type

int

#### Default value

0x80

#### Description

Number of 8x8 MACs performed per cycle (32, 64, 128, or 256).

### 3.10.28 EthosU65

Arm Ethos-U65 microNPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-948: IP revisions support**

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About EthosU65

- The EthosU65 model does not expose its registers through CADI or Iris.
- The `resetn_in` signal is active-LOW.

#### Iris and MTI instances for EthosU65

This model has the following Iris instances:

**Table 3-949: EthosU65 Iris instances**

InstanceName	ComponentName
EthosU65	EthosU65
EthosU65.pvbusmaster0	PVBusMaster
EthosU65.pvbusmaster1	PVBusMaster
EthosU65.pvbusslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-950: EthosU65 MTI instances**

InstanceName	ComponentName
EthosU65.pvbusmaster0	PVBusMaster
EthosU65.pvbusmaster1	PVBusMaster
EthosU65.pvbusslave	PVBusSlave

#### Ports for EthosU65

**Table 3-951: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	NPU clock signal
irq_out	<a href="#">Signal</a>	Master	Sends interrupt requests to the external host application processor
popl_in	<a href="#">Signal</a>	Slave	The Power-On-Reset Privilege Level (PORPL). LOW means User level. HIGH means Privileged level
posl_in	<a href="#">Signal</a>	Slave	The Power-On-Reset Security Level (PORSL). LOW means Secure. HIGH means Non-secure.
pvbus_m0	<a href="#">PVBus</a>	Master	Port 0 for NPU to access external memory
pvbus_m1	<a href="#">PVBus</a>	Master	Port 1 for NPU to access external memory

Name	Protocol	Type	Description
pvbuss_s	PVBus	Slave	Port to access NPU control registers
resetrn_in	Signal	Slave	NPU reset signal

## Parameters for EthosU65

### diagnostics

#### Type

int

#### Default value

0x0

#### Description

Enables extra information messages from the component about the NPU configuration and the PORPL/PORSL port values. Use any nonzero integer to enable all messages or 0 to disable them.

### extra\_args

#### Type

string

#### Default value

""

#### Description

To activate fast processing mode, which significantly improves the performance of the model, set this parameter to "--fast". In fast mode, NPU performance counters are not representative of counters on real hardware. We recommend you do not use any other value without instructions from Arm Technical Support.

### num\_macs

#### Type

int

#### Default value

0x100

#### Description

Number of 8x8 MACs performed per cycle (256 or 512).

## 3.10.29 EthosU85

Arm EthosU85 microNPU. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-952: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About EthosU85

- The EthosU85 model does not expose its registers through CADI or Iris.
- The `resetrn_in` signal is active-LOW.

### Iris and MTI instances for EthosU85

This model has the following Iris instances:

**Table 3-953: EthosU85 Iris instances**

InstanceName	ComponentName
EthosU85	EthosU85
EthosU85.pvbusmaster0	PVBusMaster
EthosU85.pvbusmaster1	PVBusMaster
EthosU85.pvbusslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-954: EthosU85 MTI instances**

InstanceName	ComponentName
EthosU85.pvbusmaster0	<a href="#">PVBusMaster</a>
EthosU85.pvbusmaster1	<a href="#">PVBusMaster</a>
EthosU85.pvbusslave	<a href="#">PVBusSlave</a>

### Ports for EthosU85

**Table 3-955: Ports**

Name	Protocol	Type	Description
cfgextcap_in	<a href="#">Value</a>	Slave	The configuration of capabilities for DRAM AXI ports (32 bits). Sampled with soft and hard reset.
cfgexthash0_in	<a href="#">Value_64</a>	Slave	The configuration of hash function for selecting among EXT ports (40 bits). Used to set the hash for AXI DRAM ports 0 and 1 if they are present. Sampled with soft and hard reset.
cfgsramcap_in	<a href="#">Value</a>	Slave	The configuration of capabilities for SRAM AXI ports (32 bits). Sampled with soft and hard reset.
cfgsramhash0_in	<a href="#">Value_64</a>	Slave	The configuration of hash function for selecting among SRAM ports (40 bits). Used to set the hash for AXI SRAM ports 0 and 1. Sampled with soft and hard reset.
cfgsramhash1_in	<a href="#">Value_64</a>	Slave	The configuration of hash function for selecting among SRAM ports (40 bits). Used to set the hash for AXI SRAM ports 2 and 3 if they are present. Sampled with soft and hard reset.
clk_in	<a href="#">ClockSignal</a>	Slave	NPU clock signal

Name	Protocol	Type	Description
irq_out	Signal	Master	Sends interrupt requests to the external host application processor, level triggered when HIGH.
popl_in	Signal	Slave	The Power-On-Reset Privilege Level (PORPL). LOW means User level. HIGH means Privileged level
posl_in	Signal	Slave	The Power-On-Reset Security Level (PORSL). LOW means Secure. HIGH means Non-secure.
pvbus_m0	PVBus	Master	Port 0 for NPU to access external memory
pvbus_m1	PVBus	Master	Port 1 for NPU to access external memory
pvbus_s	PVBus	Slave	Port to access NPU control registers
resetn_in	Signal	Slave	NPU reset signal (active-LOW)

## Parameters for EthosU85

### diagnostics

#### Type

int

#### Default value

0x0

#### Description

Enables extra information messages from the component about the NPU configuration and the PORPL/PORSL port values. Use any nonzero integer to enable all messages or 0 to disable them.

### extra\_args

#### Type

string

#### Default value

""

#### Description

To activate fast processing mode, which significantly improves the performance of the model, set this parameter to "--fast". In fast mode, NPU performance counters are not representative of counters on real hardware. We recommend you do not use any other value without instructions from Arm Technical Support.

### num\_macs

#### Type

int

#### Default value

0x80

#### Description

Number of 8x8 MACs performed per cycle (128, 256, 512, 1024, or 2048).

### 3.10.30 Firewall

Firewall IP. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-956: IP revisions support**

Revision	Quality level
rOp0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Iris and MTI instances for Firewall

This model has the following Iris instances:

**Table 3-957: Firewall Iris instances**

InstanceName	ComponentName
Firewall	Firewall
Firewall.BusLogger0	PVBusLogger
Firewall.BusLogger0.mapper	PVBusMapper
Firewall.BusLogger1	PVBusLogger
Firewall.BusLogger1.mapper	PVBusMapper
Firewall.BusLogger10	PVBusLogger
Firewall.BusLogger10.mapper	PVBusMapper
Firewall.BusLogger11	PVBusLogger
Firewall.BusLogger11.mapper	PVBusMapper
Firewall.BusLogger12	PVBusLogger
Firewall.BusLogger12.mapper	PVBusMapper
Firewall.BusLogger13	PVBusLogger
Firewall.BusLogger13.mapper	PVBusMapper
Firewall.BusLogger14	PVBusLogger
Firewall.BusLogger14.mapper	PVBusMapper
Firewall.BusLogger15	PVBusLogger
Firewall.BusLogger15.mapper	PVBusMapper
Firewall.BusLogger16	PVBusLogger
Firewall.BusLogger16.mapper	PVBusMapper
Firewall.BusLogger17	PVBusLogger
Firewall.BusLogger17.mapper	PVBusMapper
Firewall.BusLogger18	PVBusLogger
Firewall.BusLogger18.mapper	PVBusMapper
Firewall.BusLogger19	PVBusLogger

InstanceName	ComponentName
Firewall.BusLogger19.mapper	PVBusMapper
Firewall.BusLogger2	PVBusLogger
Firewall.BusLogger2.mapper	PVBusMapper
Firewall.BusLogger20	PVBusLogger
Firewall.BusLogger20.mapper	PVBusMapper
Firewall.BusLogger21	PVBusLogger
Firewall.BusLogger21.mapper	PVBusMapper
Firewall.BusLogger22	PVBusLogger
Firewall.BusLogger22.mapper	PVBusMapper
Firewall.BusLogger23	PVBusLogger
Firewall.BusLogger23.mapper	PVBusMapper
Firewall.BusLogger24	PVBusLogger
Firewall.BusLogger24.mapper	PVBusMapper
Firewall.BusLogger25	PVBusLogger
Firewall.BusLogger25.mapper	PVBusMapper
Firewall.BusLogger26	PVBusLogger
Firewall.BusLogger26.mapper	PVBusMapper
Firewall.BusLogger27	PVBusLogger
Firewall.BusLogger27.mapper	PVBusMapper
Firewall.BusLogger28	PVBusLogger
Firewall.BusLogger28.mapper	PVBusMapper
Firewall.BusLogger29	PVBusLogger
Firewall.BusLogger29.mapper	PVBusMapper
Firewall.BusLogger3	PVBusLogger
Firewall.BusLogger3.mapper	PVBusMapper
Firewall.BusLogger30	PVBusLogger
Firewall.BusLogger30.mapper	PVBusMapper
Firewall.BusLogger31	PVBusLogger
Firewall.BusLogger31.mapper	PVBusMapper
Firewall.BusLogger4	PVBusLogger
Firewall.BusLogger4.mapper	PVBusMapper
Firewall.BusLogger5	PVBusLogger
Firewall.BusLogger5.mapper	PVBusMapper
Firewall.BusLogger6	PVBusLogger
Firewall.BusLogger6.mapper	PVBusMapper
Firewall.BusLogger7	PVBusLogger
Firewall.BusLogger7.mapper	PVBusMapper
Firewall.BusLogger8	PVBusLogger
Firewall.BusLogger8.mapper	PVBusMapper



InstanceName	ComponentName
Firewall.BusLogger9	PVBusLogger
Firewall.BusLogger9.mapper	PVBusMapper
Firewall.BusMapper0	PVBusMapper
Firewall.BusMapper1	PVBusMapper
Firewall.BusMapper10	PVBusMapper
Firewall.BusMapper11	PVBusMapper
Firewall.BusMapper12	PVBusMapper
Firewall.BusMapper13	PVBusMapper
Firewall.BusMapper14	PVBusMapper
Firewall.BusMapper15	PVBusMapper
Firewall.BusMapper16	PVBusMapper
Firewall.BusMapper17	PVBusMapper
Firewall.BusMapper18	PVBusMapper
Firewall.BusMapper19	PVBusMapper
Firewall.BusMapper2	PVBusMapper
Firewall.BusMapper20	PVBusMapper
Firewall.BusMapper21	PVBusMapper
Firewall.BusMapper22	PVBusMapper
Firewall.BusMapper23	PVBusMapper
Firewall.BusMapper24	PVBusMapper
Firewall.BusMapper25	PVBusMapper
Firewall.BusMapper26	PVBusMapper
Firewall.BusMapper27	PVBusMapper
Firewall.BusMapper28	PVBusMapper
Firewall.BusMapper29	PVBusMapper
Firewall.BusMapper3	PVBusMapper
Firewall.BusMapper30	PVBusMapper
Firewall.BusMapper31	PVBusMapper
Firewall.BusMapper4	PVBusMapper
Firewall.BusMapper5	PVBusMapper
Firewall.BusMapper6	PVBusMapper
Firewall.BusMapper7	PVBusMapper
Firewall.BusMapper8	PVBusMapper
Firewall.BusMapper9	PVBusMapper
Firewall.bus_slave	PVBusSlave

This model has the following MTI trace components:

**Table 3-958: Firewall MTI instances**

InstanceName	ComponentName
Firewall	Firewall
Firewall.BusLogger0	PVBusLogger
Firewall.BusLogger0.mapper	PVBusMapper
Firewall.BusLogger1	PVBusLogger
Firewall.BusLogger1.mapper	PVBusMapper
Firewall.BusLogger10	PVBusLogger
Firewall.BusLogger10.mapper	PVBusMapper
Firewall.BusLogger11	PVBusLogger
Firewall.BusLogger11.mapper	PVBusMapper
Firewall.BusLogger12	PVBusLogger
Firewall.BusLogger12.mapper	PVBusMapper
Firewall.BusLogger13	PVBusLogger
Firewall.BusLogger13.mapper	PVBusMapper
Firewall.BusLogger14	PVBusLogger
Firewall.BusLogger14.mapper	PVBusMapper
Firewall.BusLogger15	PVBusLogger
Firewall.BusLogger15.mapper	PVBusMapper
Firewall.BusLogger16	PVBusLogger
Firewall.BusLogger16.mapper	PVBusMapper
Firewall.BusLogger17	PVBusLogger
Firewall.BusLogger17.mapper	PVBusMapper
Firewall.BusLogger18	PVBusLogger
Firewall.BusLogger18.mapper	PVBusMapper
Firewall.BusLogger19	PVBusLogger
Firewall.BusLogger19.mapper	PVBusMapper
Firewall.BusLogger2	PVBusLogger
Firewall.BusLogger2.mapper	PVBusMapper
Firewall.BusLogger20	PVBusLogger
Firewall.BusLogger20.mapper	PVBusMapper
Firewall.BusLogger21	PVBusLogger
Firewall.BusLogger21.mapper	PVBusMapper
Firewall.BusLogger22	PVBusLogger
Firewall.BusLogger22.mapper	PVBusMapper
Firewall.BusLogger23	PVBusLogger
Firewall.BusLogger23.mapper	PVBusMapper
Firewall.BusLogger24	PVBusLogger
Firewall.BusLogger24.mapper	PVBusMapper
Firewall.BusLogger25	PVBusLogger

InstanceName	ComponentName
Firewall.BusLogger25.mapper	PVBusMapper
Firewall.BusLogger26	PVBusLogger
Firewall.BusLogger26.mapper	PVBusMapper
Firewall.BusLogger27	PVBusLogger
Firewall.BusLogger27.mapper	PVBusMapper
Firewall.BusLogger28	PVBusLogger
Firewall.BusLogger28.mapper	PVBusMapper
Firewall.BusLogger29	PVBusLogger
Firewall.BusLogger29.mapper	PVBusMapper
Firewall.BusLogger3	PVBusLogger
Firewall.BusLogger3.mapper	PVBusMapper
Firewall.BusLogger30	PVBusLogger
Firewall.BusLogger30.mapper	PVBusMapper
Firewall.BusLogger31	PVBusLogger
Firewall.BusLogger31.mapper	PVBusMapper
Firewall.BusLogger4	PVBusLogger
Firewall.BusLogger4.mapper	PVBusMapper
Firewall.BusLogger5	PVBusLogger
Firewall.BusLogger5.mapper	PVBusMapper
Firewall.BusLogger6	PVBusLogger
Firewall.BusLogger6.mapper	PVBusMapper
Firewall.BusLogger7	PVBusLogger
Firewall.BusLogger7.mapper	PVBusMapper
Firewall.BusLogger8	PVBusLogger
Firewall.BusLogger8.mapper	PVBusMapper
Firewall.BusLogger9	PVBusLogger
Firewall.BusLogger9.mapper	PVBusMapper
Firewall.BusMapper0	PVBusMapper
Firewall.BusMapper1	PVBusMapper
Firewall.BusMapper10	PVBusMapper
Firewall.BusMapper11	PVBusMapper
Firewall.BusMapper12	PVBusMapper
Firewall.BusMapper13	PVBusMapper
Firewall.BusMapper14	PVBusMapper
Firewall.BusMapper15	PVBusMapper
Firewall.BusMapper16	PVBusMapper
Firewall.BusMapper17	PVBusMapper
Firewall.BusMapper18	PVBusMapper
Firewall.BusMapper19	PVBusMapper

InstanceName	ComponentName
Firewall.BusMapper2	PVBusMapper
Firewall.BusMapper20	PVBusMapper
Firewall.BusMapper21	PVBusMapper
Firewall.BusMapper22	PVBusMapper
Firewall.BusMapper23	PVBusMapper
Firewall.BusMapper24	PVBusMapper
Firewall.BusMapper25	PVBusMapper
Firewall.BusMapper26	PVBusMapper
Firewall.BusMapper27	PVBusMapper
Firewall.BusMapper28	PVBusMapper
Firewall.BusMapper29	PVBusMapper
Firewall.BusMapper3	PVBusMapper
Firewall.BusMapper30	PVBusMapper
Firewall.BusMapper31	PVBusMapper
Firewall.BusMapper4	PVBusMapper
Firewall.BusMapper5	PVBusMapper
Firewall.BusMapper6	PVBusMapper
Firewall.BusMapper7	PVBusMapper
Firewall.BusMapper8	PVBusMapper
Firewall.BusMapper9	PVBusMapper
Firewall.bus_slave	PVBusSlave

## Ports for Firewall

**Table 3-959: Ports**

Name	Protocol	Type	Description
irq_signal	Signal	Master	-
irq_signal_tamper	Signal	Master	-
lockdown	Signal	Slave	-
pvbus_component_m[31]	PVBus	Master	-
pvbus_component_s[31]	PVBus	Slave	-
pvbus_program_iface	PVBus	Slave	-
reset_signal	Signal	Slave	-

## Parameters for Firewall

### ADDR\_WIDTH

#### Type

string















To use the GIC500 component, you must configure some parameters. For example:

```
gic500: GIC500(
    "num_clusters" = 2,
    "cpus_per_cluster_0" = 4,
    "cpus_per_cluster_1" = 4,
    "reg-base" = 0x2c200000,
    "SPI-count" = 256
);
```



Note

- To print to stderr the memory map of any GICv3 or later models that are included in the platform, set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1.
- For writes to `GITS_TRANSLATE64R`, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see the [MSIRewriter](#) component.

## MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-961: MasterID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



Note

The `pvbus_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

## Iris and MTI instances for GIC500

This model has the following Iris instances:

**Table 3-962: GIC500 Iris instances**

InstanceName	ComponentName
GIC500	GIC_IRI
GIC500.ITS0	GICv3InterruptTranslationService
GIC500.rd_0	GICv3RedistributorInternal
GIC500.rd_0_0	GICv3RedistributorInternal
GIC500.rd_0_0_0	GICv3RedistributorInternal
GIC500.rd_0_0_0_0	GICv3Redistributor
GIC500.rd_tl	GICv3Distributor

This model has the following MTI trace components:

**Table 3-963: GIC500 MTI instances**

InstanceName	ComponentName
GIC500.ITS0	GICv3InterruptTranslationService
GIC500.rd_0	GICv3RedistributorInternal
GIC500.rd_0_0	GICv3RedistributorInternal
GIC500.rd_0_0_0	GICv3RedistributorInternal
GIC500.rd_0_0_0_0	GICv3Redistributor
GIC500.rd_t1	GICv3Distributor

## Ports for GIC500

**Table 3-964: Ports**

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Disable some SPIs signal.
cpu_active_0[8]	Signal	Slave	cpu_active pins of cluster 0.
cpu_active_1[8]	Signal	Slave	cpu_active pins of cluster 1.
cpu_active_10[8]	Signal	Slave	cpu_active pins of cluster 10.
cpu_active_11[8]	Signal	Slave	cpu_active pins of cluster 11.
cpu_active_12[8]	Signal	Slave	cpu_active pins of cluster 12.
cpu_active_13[8]	Signal	Slave	cpu_active pins of cluster 13.
cpu_active_14[8]	Signal	Slave	cpu_active pins of cluster 14.
cpu_active_15[8]	Signal	Slave	cpu_active pins of cluster 15.
cpu_active_16[8]	Signal	Slave	cpu_active pins of cluster 16.
cpu_active_17[8]	Signal	Slave	cpu_active pins of cluster 17.
cpu_active_18[8]	Signal	Slave	cpu_active pins of cluster 18.
cpu_active_19[8]	Signal	Slave	cpu_active pins of cluster 19.
cpu_active_2[8]	Signal	Slave	cpu_active pins of cluster 2.
cpu_active_20[8]	Signal	Slave	cpu_active pins of cluster 20.
cpu_active_21[8]	Signal	Slave	cpu_active pins of cluster 21.
cpu_active_22[8]	Signal	Slave	cpu_active pins of cluster 22.
cpu_active_23[8]	Signal	Slave	cpu_active pins of cluster 23.
cpu_active_24[8]	Signal	Slave	cpu_active pins of cluster 24.
cpu_active_25[8]	Signal	Slave	cpu_active pins of cluster 25.
cpu_active_26[8]	Signal	Slave	cpu_active pins of cluster 26.
cpu_active_27[8]	Signal	Slave	cpu_active pins of cluster 27.
cpu_active_28[8]	Signal	Slave	cpu_active pins of cluster 28.
cpu_active_29[8]	Signal	Slave	cpu_active pins of cluster 29.
cpu_active_3[8]	Signal	Slave	cpu_active pins of cluster 3.
cpu_active_30[8]	Signal	Slave	cpu_active pins of cluster 30.

Name	Protocol	Type	Description
cpu_active_31[8]	Signal	Slave	cpu_active pins of cluster 31.
cpu_active_4[8]	Signal	Slave	cpu_active pins of cluster 4.
cpu_active_5[8]	Signal	Slave	cpu_active pins of cluster 5.
cpu_active_6[8]	Signal	Slave	cpu_active pins of cluster 6.
cpu_active_7[8]	Signal	Slave	cpu_active pins of cluster 7.
cpu_active_8[8]	Signal	Slave	cpu_active pins of cluster 8.
cpu_active_9[8]	Signal	Slave	cpu_active pins of cluster 9.
po_reset	Signal	Slave	Power on reset.
ppi16_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 0.
ppi16_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 1.
ppi16_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 10.
ppi16_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 11.
ppi16_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 12.
ppi16_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 13.
ppi16_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 14.
ppi16_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 15.
ppi16_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 16.
ppi16_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 17.
ppi16_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 18.
ppi16_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 19.
ppi16_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 2.
ppi16_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 20.
ppi16_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 21.
ppi16_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 22.
ppi16_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 23.
ppi16_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 24.
ppi16_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 25.
ppi16_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 26.
ppi16_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 27.
ppi16_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 28.
ppi16_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 29.
ppi16_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 3.
ppi16_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 30.
ppi16_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 31.
ppi16_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 4.
ppi16_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 5.
ppi16_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 6.
ppi16_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 7.
ppi16_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 8.

Name	Protocol	Type	Description
ppi16_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 9.
ppi17_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 0.
ppi17_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 1.
ppi17_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 10.
ppi17_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 11.
ppi17_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 12.
ppi17_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 13.
ppi17_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 14.
ppi17_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 15.
ppi17_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 16.
ppi17_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 17.
ppi17_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 18.
ppi17_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 19.
ppi17_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 2.
ppi17_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 20.
ppi17_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 21.
ppi17_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 22.
ppi17_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 23.
ppi17_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 24.
ppi17_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 25.
ppi17_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 26.
ppi17_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 27.
ppi17_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 28.
ppi17_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 29.
ppi17_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 3.
ppi17_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 30.
ppi17_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 31.
ppi17_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 4.
ppi17_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 5.
ppi17_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 6.
ppi17_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 7.
ppi17_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 8.
ppi17_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 9.
ppi18_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 0.
ppi18_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 1.
ppi18_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 10.
ppi18_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 11.
ppi18_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 12.
ppi18_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 13.

Name	Protocol	Type	Description
ppi18_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 14.
ppi18_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 15.
ppi18_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 16.
ppi18_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 17.
ppi18_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 18.
ppi18_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 19.
ppi18_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 2.
ppi18_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 20.
ppi18_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 21.
ppi18_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 22.
ppi18_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 23.
ppi18_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 24.
ppi18_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 25.
ppi18_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 26.
ppi18_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 27.
ppi18_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 28.
ppi18_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 29.
ppi18_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 3.
ppi18_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 30.
ppi18_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 31.
ppi18_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 4.
ppi18_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 5.
ppi18_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 6.
ppi18_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 7.
ppi18_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 8.
ppi18_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 9.
ppi19_in_0[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 0.
ppi19_in_1[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 1.
ppi19_in_10[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 10.
ppi19_in_11[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 11.
ppi19_in_12[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 12.
ppi19_in_13[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 13.
ppi19_in_14[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 14.
ppi19_in_15[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 15.
ppi19_in_16[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 16.
ppi19_in_17[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 17.
ppi19_in_18[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 18.
ppi19_in_19[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 19.
ppi19_in_2[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 2.

Name	Protocol	Type	Description
ppi19_in_20[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 20.
ppi19_in_21[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 21.
ppi19_in_22[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 22.
ppi19_in_23[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 23.
ppi19_in_24[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 24.
ppi19_in_25[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 25.
ppi19_in_26[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 26.
ppi19_in_27[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 27.
ppi19_in_28[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 28.
ppi19_in_29[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 29.
ppi19_in_3[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 3.
ppi19_in_30[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 30.
ppi19_in_31[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 31.
ppi19_in_4[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 4.
ppi19_in_5[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 5.
ppi19_in_6[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 6.
ppi19_in_7[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 7.
ppi19_in_8[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 8.
ppi19_in_9[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 9.
ppi20_in_0[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 0.
ppi20_in_1[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 1.
ppi20_in_10[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 10.
ppi20_in_11[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 11.
ppi20_in_12[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 12.
ppi20_in_13[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 13.
ppi20_in_14[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 14.
ppi20_in_15[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 15.
ppi20_in_16[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 16.
ppi20_in_17[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 17.
ppi20_in_18[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 18.
ppi20_in_19[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 19.
ppi20_in_2[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 2.
ppi20_in_20[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 20.
ppi20_in_21[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 21.
ppi20_in_22[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 22.
ppi20_in_23[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 23.
ppi20_in_24[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 24.
ppi20_in_25[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 25.
ppi20_in_26[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 26.



Name	Protocol	Type	Description
ppi20_in_27[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 27.
ppi20_in_28[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 28.
ppi20_in_29[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 29.
ppi20_in_3[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 3.
ppi20_in_30[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 30.
ppi20_in_31[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 31.
ppi20_in_4[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 4.
ppi20_in_5[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 5.
ppi20_in_6[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 6.
ppi20_in_7[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 7.
ppi20_in_8[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 8.
ppi20_in_9[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 9.
ppi21_in_0[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 0.
ppi21_in_1[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 1.
ppi21_in_10[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 10.
ppi21_in_11[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 11.
ppi21_in_12[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 12.
ppi21_in_13[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 13.
ppi21_in_14[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 14.
ppi21_in_15[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 15.
ppi21_in_16[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 16.
ppi21_in_17[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 17.
ppi21_in_18[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 18.
ppi21_in_19[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 19.
ppi21_in_2[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 2.
ppi21_in_20[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 20.
ppi21_in_21[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 21.
ppi21_in_22[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 22.
ppi21_in_23[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 23.
ppi21_in_24[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 24.
ppi21_in_25[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 25.
ppi21_in_26[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 26.
ppi21_in_27[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 27.
ppi21_in_28[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 28.
ppi21_in_29[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 29.
ppi21_in_3[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 3.
ppi21_in_30[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 30.
ppi21_in_31[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 31.
ppi21_in_4[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 4.

Name	Protocol	Type	Description
ppi21_in_5[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 5.
ppi21_in_6[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 6.
ppi21_in_7[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 7.
ppi21_in_8[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 8.
ppi21_in_9[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 9.
ppi22_in_0[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 0.
ppi22_in_1[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 1.
ppi22_in_10[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 10.
ppi22_in_11[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 11.
ppi22_in_12[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 12.
ppi22_in_13[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 13.
ppi22_in_14[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 14.
ppi22_in_15[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 15.
ppi22_in_16[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 16.
ppi22_in_17[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 17.
ppi22_in_18[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 18.
ppi22_in_19[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 19.
ppi22_in_2[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 2.
ppi22_in_20[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 20.
ppi22_in_21[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 21.
ppi22_in_22[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 22.
ppi22_in_23[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 23.
ppi22_in_24[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 24.
ppi22_in_25[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 25.
ppi22_in_26[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 26.

Name	Protocol	Type	Description
ppi22_in_27[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 27.
ppi22_in_28[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 28.
ppi22_in_29[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 29.
ppi22_in_3[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 3.
ppi22_in_30[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 30.
ppi22_in_31[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 31.
ppi22_in_4[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 4.
ppi22_in_5[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 5.
ppi22_in_6[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 6.
ppi22_in_7[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 7.
ppi22_in_8[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 8.
ppi22_in_9[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 9.
ppi23_in_0[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 0.
ppi23_in_1[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 1.
ppi23_in_10[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 10.
ppi23_in_11[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 11.
ppi23_in_12[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 12.
ppi23_in_13[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 13.
ppi23_in_14[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 14.
ppi23_in_15[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 15.
ppi23_in_16[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 16.
ppi23_in_17[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 17.
ppi23_in_18[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 18.

Name	Protocol	Type	Description
ppi23_in_19[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 19.
ppi23_in_2[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 2.
ppi23_in_20[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 20.
ppi23_in_21[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 21.
ppi23_in_22[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 22.
ppi23_in_23[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 23.
ppi23_in_24[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 24.
ppi23_in_25[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 25.
ppi23_in_26[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 26.
ppi23_in_27[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 27.
ppi23_in_28[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 28.
ppi23_in_29[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 29.
ppi23_in_3[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 3.
ppi23_in_30[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 30.
ppi23_in_31[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 31.
ppi23_in_4[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 4.
ppi23_in_5[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 5.
ppi23_in_6[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 6.
ppi23_in_7[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 7.
ppi23_in_8[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 8.
ppi23_in_9[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 9.
ppi24_in_0[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 0.
ppi24_in_1[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 1.

Name	Protocol	Type	Description
ppi24_in_10[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 10.
ppi24_in_11[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 11.
ppi24_in_12[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 12.
ppi24_in_13[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 13.
ppi24_in_14[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 14.
ppi24_in_15[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 15.
ppi24_in_16[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 16.
ppi24_in_17[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 17.
ppi24_in_18[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 18.
ppi24_in_19[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 19.
ppi24_in_2[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 2.
ppi24_in_20[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 20.
ppi24_in_21[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 21.
ppi24_in_22[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 22.
ppi24_in_23[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 23.
ppi24_in_24[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 24.
ppi24_in_25[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 25.
ppi24_in_26[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 26.
ppi24_in_27[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 27.
ppi24_in_28[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 28.
ppi24_in_29[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 29.
ppi24_in_3[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 3.
ppi24_in_30[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 30.

Name	Protocol	Type	Description
ppi24_in_31[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 31.
ppi24_in_4[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 4.
ppi24_in_5[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 5.
ppi24_in_6[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 6.
ppi24_in_7[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 7.
ppi24_in_8[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 8.
ppi24_in_9[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 9.
ppi25_in_0[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 0.
ppi25_in_1[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 1.
ppi25_in_10[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 10.
ppi25_in_11[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 11.
ppi25_in_12[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 12.
ppi25_in_13[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 13.
ppi25_in_14[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 14.
ppi25_in_15[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 15.
ppi25_in_16[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 16.
ppi25_in_17[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 17.
ppi25_in_18[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 18.
ppi25_in_19[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 19.
ppi25_in_2[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 2.
ppi25_in_20[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 20.
ppi25_in_21[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 21.
ppi25_in_22[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 22.

Name	Protocol	Type	Description
ppi25_in_23[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 23.
ppi25_in_24[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 24.
ppi25_in_25[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 25.
ppi25_in_26[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 26.
ppi25_in_27[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 27.
ppi25_in_28[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 28.
ppi25_in_29[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 29.
ppi25_in_3[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 3.
ppi25_in_30[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 30.
ppi25_in_31[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 31.
ppi25_in_4[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 4.
ppi25_in_5[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 5.
ppi25_in_6[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 6.
ppi25_in_7[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 7.
ppi25_in_8[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 8.
ppi25_in_9[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 9.
ppi26_in_0[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 0.
ppi26_in_1[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 1.
ppi26_in_10[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 10.
ppi26_in_11[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 11.
ppi26_in_12[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 12.
ppi26_in_13[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 13.
ppi26_in_14[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 14.
ppi26_in_15[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 15.
ppi26_in_16[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 16.
ppi26_in_17[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 17.
ppi26_in_18[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 18.
ppi26_in_19[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 19.



Name	Protocol	Type	Description
ppi26_in_2[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 2.
ppi26_in_20[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 20.
ppi26_in_21[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 21.
ppi26_in_22[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 22.
ppi26_in_23[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 23.
ppi26_in_24[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 24.
ppi26_in_25[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 25.
ppi26_in_26[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 26.
ppi26_in_27[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 27.
ppi26_in_28[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 28.
ppi26_in_29[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 29.
ppi26_in_3[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 3.
ppi26_in_30[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 30.
ppi26_in_31[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 31.
ppi26_in_4[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 4.
ppi26_in_5[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 5.
ppi26_in_6[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 6.
ppi26_in_7[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 7.
ppi26_in_8[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 8.
ppi26_in_9[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 9.
ppi27_in_0[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 0.
ppi27_in_1[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 1.
ppi27_in_10[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 10.
ppi27_in_11[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 11.
ppi27_in_12[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 12.
ppi27_in_13[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 13.
ppi27_in_14[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 14.
ppi27_in_15[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 15.
ppi27_in_16[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 16.
ppi27_in_17[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 17.
ppi27_in_18[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 18.
ppi27_in_19[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 19.
ppi27_in_2[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 2.
ppi27_in_20[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 20.
ppi27_in_21[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 21.
ppi27_in_22[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 22.
ppi27_in_23[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 23.
ppi27_in_24[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 24.
ppi27_in_25[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 25.



Name	Protocol	Type	Description
ppi27_in_26[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 26.
ppi27_in_27[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 27.
ppi27_in_28[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 28.
ppi27_in_29[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 29.
ppi27_in_3[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 3.
ppi27_in_30[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 30.
ppi27_in_31[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 31.
ppi27_in_4[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 4.
ppi27_in_5[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 5.
ppi27_in_6[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 6.
ppi27_in_7[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 7.
ppi27_in_8[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 8.
ppi27_in_9[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 9.
ppi28_in_0[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 0.
ppi28_in_1[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 1.
ppi28_in_10[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 10.
ppi28_in_11[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 11.
ppi28_in_12[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 12.
ppi28_in_13[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 13.
ppi28_in_14[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 14.
ppi28_in_15[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 15.
ppi28_in_16[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 16.
ppi28_in_17[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 17.
ppi28_in_18[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 18.
ppi28_in_19[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 19.
ppi28_in_2[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 2.
ppi28_in_20[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 20.
ppi28_in_21[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 21.
ppi28_in_22[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 22.
ppi28_in_23[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 23.
ppi28_in_24[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 24.
ppi28_in_25[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 25.
ppi28_in_26[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 26.
ppi28_in_27[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 27.
ppi28_in_28[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 28.
ppi28_in_29[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 29.
ppi28_in_3[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 3.
ppi28_in_30[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 30.
ppi28_in_31[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 31.

Name	Protocol	Type	Description
ppi28_in_4[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 4.
ppi28_in_5[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 5.
ppi28_in_6[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 6.
ppi28_in_7[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 7.
ppi28_in_8[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 8.
ppi28_in_9[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 9.
ppi29_in_0[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 0.
ppi29_in_1[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 1.
ppi29_in_10[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 10.
ppi29_in_11[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 11.
ppi29_in_12[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 12.
ppi29_in_13[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 13.
ppi29_in_14[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 14.
ppi29_in_15[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 15.
ppi29_in_16[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 16.
ppi29_in_17[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 17.
ppi29_in_18[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 18.
ppi29_in_19[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 19.
ppi29_in_2[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 2.
ppi29_in_20[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 20.
ppi29_in_21[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 21.
ppi29_in_22[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 22.
ppi29_in_23[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 23.
ppi29_in_24[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 24.
ppi29_in_25[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 25.
ppi29_in_26[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 26.
ppi29_in_27[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 27.
ppi29_in_28[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 28.
ppi29_in_29[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 29.
ppi29_in_3[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 3.
ppi29_in_30[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 30.
ppi29_in_31[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 31.
ppi29_in_4[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 4.
ppi29_in_5[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 5.
ppi29_in_6[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 6.
ppi29_in_7[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 7.
ppi29_in_8[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 8.
ppi29_in_9[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 9.

Name	Protocol	Type	Description
ppi30_in_0[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 0.
ppi30_in_1[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 1.
ppi30_in_10[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 10.
ppi30_in_11[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 11.
ppi30_in_12[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 12.
ppi30_in_13[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 13.
ppi30_in_14[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 14.
ppi30_in_15[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 15.
ppi30_in_16[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 16.
ppi30_in_17[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 17.
ppi30_in_18[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 18.
ppi30_in_19[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 19.
ppi30_in_2[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 2.
ppi30_in_20[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 20.
ppi30_in_21[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 21.
ppi30_in_22[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 22.
ppi30_in_23[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 23.
ppi30_in_24[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 24.
ppi30_in_25[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 25.
ppi30_in_26[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 26.
ppi30_in_27[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 27.
ppi30_in_28[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 28.
ppi30_in_29[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 29.

Name	Protocol	Type	Description
ppi30_in_3[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 3.
ppi30_in_30[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 30.
ppi30_in_31[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 31.
ppi30_in_4[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 4.
ppi30_in_5[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 5.
ppi30_in_6[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 6.
ppi30_in_7[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 7.
ppi30_in_8[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 8.
ppi30_in_9[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 9.
ppi31_in_0[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 0.
ppi31_in_1[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 1.
ppi31_in_10[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 10.
ppi31_in_11[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 11.
ppi31_in_12[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 12.
ppi31_in_13[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 13.
ppi31_in_14[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 14.
ppi31_in_15[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 15.
ppi31_in_16[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 16.
ppi31_in_17[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 17.
ppi31_in_18[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 18.
ppi31_in_19[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 19.
ppi31_in_2[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 2.
ppi31_in_20[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 20.
ppi31_in_21[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 21.
ppi31_in_22[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 22.
ppi31_in_23[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 23.
ppi31_in_24[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 24.
ppi31_in_25[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 25.
ppi31_in_26[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 26.
ppi31_in_27[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 27.
ppi31_in_28[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 28.
ppi31_in_29[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 29.
ppi31_in_3[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 3.

Name	Protocol	Type	Description
ppi31_in_30[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 30.
ppi31_in_31[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 31.
ppi31_in_4[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 4.
ppi31_in_5[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 5.
ppi31_in_6[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 6.
ppi31_in_7[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 7.
ppi31_in_8[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 8.
ppi31_in_9[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 9.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Reset.
spi_in[988]	Signal	Slave	Shared peripheral interrupts.
wake_request_0[8]	Signal	Master	Power management outputs of cluster 0.
wake_request_1[8]	Signal	Master	Power management outputs of cluster 1.
wake_request_10[8]	Signal	Master	Power management outputs of cluster 10.
wake_request_11[8]	Signal	Master	Power management outputs of cluster 11.
wake_request_12[8]	Signal	Master	Power management outputs of cluster 12.
wake_request_13[8]	Signal	Master	Power management outputs of cluster 13.
wake_request_14[8]	Signal	Master	Power management outputs of cluster 14.
wake_request_15[8]	Signal	Master	Power management outputs of cluster 15.
wake_request_16[8]	Signal	Master	Power management outputs of cluster 16.
wake_request_17[8]	Signal	Master	Power management outputs of cluster 17.
wake_request_18[8]	Signal	Master	Power management outputs of cluster 18.
wake_request_19[8]	Signal	Master	Power management outputs of cluster 19.
wake_request_2[8]	Signal	Master	Power management outputs of cluster 2.
wake_request_20[8]	Signal	Master	Power management outputs of cluster 20.
wake_request_21[8]	Signal	Master	Power management outputs of cluster 21.
wake_request_22[8]	Signal	Master	Power management outputs of cluster 22.
wake_request_23[8]	Signal	Master	Power management outputs of cluster 23.
wake_request_24[8]	Signal	Master	Power management outputs of cluster 24.
wake_request_25[8]	Signal	Master	Power management outputs of cluster 25.
wake_request_26[8]	Signal	Master	Power management outputs of cluster 26.
wake_request_27[8]	Signal	Master	Power management outputs of cluster 27.
wake_request_28[8]	Signal	Master	Power management outputs of cluster 28.
wake_request_29[8]	Signal	Master	Power management outputs of cluster 29.
wake_request_3[8]	Signal	Master	Power management outputs of cluster 3.
wake_request_30[8]	Signal	Master	Power management outputs of cluster 30.
wake_request_31[8]	Signal	Master	Power management outputs of cluster 31.

Name	Protocol	Type	Description
wake_request_4[8]	Signal	Master	Power management outputs of cluster 4.
wake_request_5[8]	Signal	Master	Power management outputs of cluster 5.
wake_request_6[8]	Signal	Master	Power management outputs of cluster 6.
wake_request_7[8]	Signal	Master	Power management outputs of cluster 7.
wake_request_8[8]	Signal	Master	Power management outputs of cluster 8.
wake_request_9[8]	Signal	Master	Power management outputs of cluster 9.

## Parameters for GIC500

### GICD\_ITARGETSR-RAZWI

#### Type

bool

#### Default value

0x0

#### Description

If true, the GICD\_ITARGETS registers are RAZ/WI.

### ITS-count

#### Type

int

#### Default value

0x1

#### Description

Number of Interrupt Translation Services to be instantiated (0=none).

### ITS-device-bits

#### Type

int

#### Default value

0x10

#### Description

Number of bits supported for ITS device IDs.

### ITS-threaded-command-queue

#### Type

bool

#### Default value

0x1

**Description**

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

**SPI-count****Type**

int

**Default value**

0xe0

**Description**

Number of SPIs that are implemented.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**cpus\_per\_cluster\_0****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 0.

**cpus\_per\_cluster\_1****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 1.

**cpus\_per\_cluster\_10****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 10.

**cpus\_per\_cluster\_11****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 11.

**cpus\_per\_cluster\_12****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 12.

**cpus\_per\_cluster\_13****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 13.

**cpus\_per\_cluster\_14****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 14.

**cpus\_per\_cluster\_15****Type**

int



**Default value**

0x1

**Description**

Number of cores within cluster 15.

**cpus\_per\_cluster\_16****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 16.

**cpus\_per\_cluster\_17****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 17.

**cpus\_per\_cluster\_18****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 18.

**cpus\_per\_cluster\_19****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 19.

**cpus\_per\_cluster\_2****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 2.

**cpus\_per\_cluster\_20****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 20.

**cpus\_per\_cluster\_21****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 21.

**cpus\_per\_cluster\_22****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 22.

**cpus\_per\_cluster\_23****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 23.

**cpus\_per\_cluster\_24****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 24.

**cpus\_per\_cluster\_25****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 25.

**cpus\_per\_cluster\_26****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 26.

**cpus\_per\_cluster\_27****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 27.

**cpus\_per\_cluster\_28****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 28.

**cpus\_per\_cluster\_29****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 29.

**cpus\_per\_cluster\_3****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 3.

**cpus\_per\_cluster\_30****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 30.

**cpus\_per\_cluster\_31****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 31.

**cpus\_per\_cluster\_4****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 4.

**cpus\_per\_cluster\_5****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 5.

**cpus\_per\_cluster\_6****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 6.

**cpus\_per\_cluster\_7****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 7.

**cpus\_per\_cluster\_8****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 8.

**cpus\_per\_cluster\_9****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 9.

**delay-ITS-accesses****Type**

bool

**Default value**

0x1

**Description**

Delay accesses from the ITS until GICR\_SYNCRR is read.

**delay-redistributor-accesses****Type**

bool

**Default value**

0x1

**Description**

Delay memory accesses from the redistributor until GICR\_SYNCRR is read.

**enable\_protocol\_checking****Type**

bool

**Default value**

0x0

**Description**

Enable/disable protocol checking at cpu interface.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**has-two-security-states****Type**

bool

**Default value**

0x1

**Description**

If true, has two security states.

**num\_clusters****Type**

int

**Default value**

0x1

**Description**

Number of implemented affinity level1 clusters.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**redistributor-threaded-sync****Type**

bool

**Default value**

0x1

**Description**

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

**reg-base****Type**

int

**Default value**

0x10000000

**Description**

GIC500 base address.

**using-generated-memorymap****Type**

bool

**Default value**

0x1

**Description**

Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.

**wakeup-on-reset****Type**

bool

**Default value**

0x0

**Description**

Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

### 3.10.32 GIC500\_ClusterPorts

GIC500 Component for distribution of interrupts. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-965: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About GIC500\_ClusterPorts**

An alternate version of GIC500, identical to the above except for the CPUIF ports being exposed as an array per cluster

**Iris and MTI instances for GIC500\_ClusterPorts**

This model has the following Iris instances:

**Table 3-966: GIC500\_ClusterPorts Iris instances**

InstanceName	ComponentName
GIC500_ClusterPorts	GIC_IRI
GIC500_ClusterPorts.ITS0	GICv3InterruptTranslationService
GIC500_ClusterPorts.rd_0	GICv3RedistributorInternal
GIC500_ClusterPorts.rd_0_0	GICv3RedistributorInternal
GIC500_ClusterPorts.rd_0_0_0	GICv3RedistributorInternal
GIC500_ClusterPorts.rd_0_0_0_0	GICv3Redistributor
GIC500_ClusterPorts.rd_tl	GICv3Distributor

This model has the following MTI trace components:



**Table 3-967: GIC500\_ClusterPorts MTI instances**

InstanceName	ComponentName
GIC500_ClusterPorts.ITS0	GICv3InterruptTranslationService
GIC500_ClusterPorts.rd_0	GICv3RedistributorInternal
GIC500_ClusterPorts.rd_0_0	GICv3RedistributorInternal
GIC500_ClusterPorts.rd_0_0_0	GICv3RedistributorInternal
GIC500_ClusterPorts.rd_0_0_0_0	GICv3Redistributor
GIC500_ClusterPorts.rd_tl	GICv3Distributor

**Ports for GIC500\_ClusterPorts****Table 3-968: Ports**

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Disable some SPLs signal.
cpu_active_0[8]	Signal	Slave	cpu_active pins of cluster 0.
cpu_active_1[8]	Signal	Slave	cpu_active pins of cluster 1.
cpu_active_10[8]	Signal	Slave	cpu_active pins of cluster 10.
cpu_active_11[8]	Signal	Slave	cpu_active pins of cluster 11.
cpu_active_12[8]	Signal	Slave	cpu_active pins of cluster 12.
cpu_active_13[8]	Signal	Slave	cpu_active pins of cluster 13.
cpu_active_14[8]	Signal	Slave	cpu_active pins of cluster 14.
cpu_active_15[8]	Signal	Slave	cpu_active pins of cluster 15.
cpu_active_16[8]	Signal	Slave	cpu_active pins of cluster 16.
cpu_active_17[8]	Signal	Slave	cpu_active pins of cluster 17.
cpu_active_18[8]	Signal	Slave	cpu_active pins of cluster 18.
cpu_active_19[8]	Signal	Slave	cpu_active pins of cluster 19.
cpu_active_2[8]	Signal	Slave	cpu_active pins of cluster 2.
cpu_active_20[8]	Signal	Slave	cpu_active pins of cluster 20.
cpu_active_21[8]	Signal	Slave	cpu_active pins of cluster 21.
cpu_active_22[8]	Signal	Slave	cpu_active pins of cluster 22.
cpu_active_23[8]	Signal	Slave	cpu_active pins of cluster 23.
cpu_active_24[8]	Signal	Slave	cpu_active pins of cluster 24.
cpu_active_25[8]	Signal	Slave	cpu_active pins of cluster 25.
cpu_active_26[8]	Signal	Slave	cpu_active pins of cluster 26.
cpu_active_27[8]	Signal	Slave	cpu_active pins of cluster 27.
cpu_active_28[8]	Signal	Slave	cpu_active pins of cluster 28.
cpu_active_29[8]	Signal	Slave	cpu_active pins of cluster 29.
cpu_active_3[8]	Signal	Slave	cpu_active pins of cluster 3.
cpu_active_30[8]	Signal	Slave	cpu_active pins of cluster 30.
cpu_active_31[8]	Signal	Slave	cpu_active pins of cluster 31.
cpu_active_4[8]	Signal	Slave	cpu_active pins of cluster 4.

Name	Protocol	Type	Description
cpu_active_5[8]	Signal	Slave	cpu_active pins of cluster 5.
cpu_active_6[8]	Signal	Slave	cpu_active pins of cluster 6.
cpu_active_7[8]	Signal	Slave	cpu_active pins of cluster 7.
cpu_active_8[8]	Signal	Slave	cpu_active pins of cluster 8.
cpu_active_9[8]	Signal	Slave	cpu_active pins of cluster 9.
po_reset	Signal	Slave	Power on reset.
ppi16_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 0.
ppi16_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 1.
ppi16_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 10.
ppi16_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 11.
ppi16_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 12.
ppi16_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 13.
ppi16_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 14.
ppi16_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 15.
ppi16_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 16.
ppi16_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 17.
ppi16_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 18.
ppi16_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 19.
ppi16_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 2.
ppi16_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 20.
ppi16_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 21.
ppi16_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 22.
ppi16_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 23.
ppi16_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 24.
ppi16_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 25.
ppi16_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 26.
ppi16_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 27.
ppi16_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 28.
ppi16_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 29.
ppi16_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 3.
ppi16_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 30.
ppi16_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 31.
ppi16_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 4.
ppi16_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 5.
ppi16_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 6.
ppi16_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 7.
ppi16_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 8.
ppi16_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 9.
ppi17_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 0.

Name	Protocol	Type	Description
ppi17_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 1.
ppi17_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 10.
ppi17_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 11.
ppi17_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 12.
ppi17_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 13.
ppi17_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 14.
ppi17_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 15.
ppi17_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 16.
ppi17_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 17.
ppi17_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 18.
ppi17_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 19.
ppi17_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 2.
ppi17_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 20.
ppi17_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 21.
ppi17_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 22.
ppi17_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 23.
ppi17_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 24.
ppi17_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 25.
ppi17_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 26.
ppi17_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 27.
ppi17_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 28.
ppi17_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 29.
ppi17_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 3.
ppi17_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 30.
ppi17_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 31.
ppi17_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 4.
ppi17_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 5.
ppi17_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 6.
ppi17_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 7.
ppi17_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 8.
ppi17_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 9.
ppi18_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 0.
ppi18_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 1.
ppi18_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 10.
ppi18_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 11.
ppi18_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 12.
ppi18_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 13.
ppi18_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 14.
ppi18_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 15.

Name	Protocol	Type	Description
ppi18_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 16.
ppi18_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 17.
ppi18_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 18.
ppi18_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 19.
ppi18_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 2.
ppi18_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 20.
ppi18_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 21.
ppi18_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 22.
ppi18_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 23.
ppi18_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 24.
ppi18_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 25.
ppi18_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 26.
ppi18_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 27.
ppi18_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 28.
ppi18_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 29.
ppi18_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 3.
ppi18_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 30.
ppi18_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 31.
ppi18_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 4.
ppi18_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 5.
ppi18_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 6.
ppi18_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 7.
ppi18_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 8.
ppi18_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 9.
ppi19_in_0[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 0.
ppi19_in_1[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 1.
ppi19_in_10[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 10.
ppi19_in_11[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 11.
ppi19_in_12[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 12.
ppi19_in_13[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 13.
ppi19_in_14[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 14.
ppi19_in_15[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 15.
ppi19_in_16[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 16.
ppi19_in_17[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 17.
ppi19_in_18[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 18.
ppi19_in_19[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 19.
ppi19_in_2[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 2.
ppi19_in_20[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 20.
ppi19_in_21[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 21.

Name	Protocol	Type	Description
ppi19_in_22[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 22.
ppi19_in_23[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 23.
ppi19_in_24[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 24.
ppi19_in_25[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 25.
ppi19_in_26[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 26.
ppi19_in_27[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 27.
ppi19_in_28[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 28.
ppi19_in_29[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 29.
ppi19_in_3[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 3.
ppi19_in_30[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 30.
ppi19_in_31[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 31.
ppi19_in_4[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 4.
ppi19_in_5[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 5.
ppi19_in_6[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 6.
ppi19_in_7[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 7.
ppi19_in_8[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 8.
ppi19_in_9[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 9.
ppi20_in_0[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 0.
ppi20_in_1[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 1.
ppi20_in_10[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 10.
ppi20_in_11[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 11.
ppi20_in_12[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 12.
ppi20_in_13[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 13.
ppi20_in_14[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 14.
ppi20_in_15[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 15.
ppi20_in_16[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 16.
ppi20_in_17[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 17.
ppi20_in_18[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 18.
ppi20_in_19[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 19.
ppi20_in_2[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 2.
ppi20_in_20[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 20.
ppi20_in_21[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 21.
ppi20_in_22[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 22.
ppi20_in_23[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 23.
ppi20_in_24[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 24.
ppi20_in_25[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 25.
ppi20_in_26[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 26.
ppi20_in_27[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 27.
ppi20_in_28[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 28.

Name	Protocol	Type	Description
ppi20_in_29[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 29.
ppi20_in_3[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 3.
ppi20_in_30[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 30.
ppi20_in_31[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 31.
ppi20_in_4[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 4.
ppi20_in_5[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 5.
ppi20_in_6[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 6.
ppi20_in_7[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 7.
ppi20_in_8[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 8.
ppi20_in_9[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 9.
ppi21_in_0[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 0.
ppi21_in_1[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 1.
ppi21_in_10[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 10.
ppi21_in_11[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 11.
ppi21_in_12[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 12.
ppi21_in_13[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 13.
ppi21_in_14[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 14.
ppi21_in_15[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 15.
ppi21_in_16[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 16.
ppi21_in_17[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 17.
ppi21_in_18[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 18.
ppi21_in_19[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 19.
ppi21_in_2[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 2.
ppi21_in_20[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 20.
ppi21_in_21[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 21.
ppi21_in_22[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 22.
ppi21_in_23[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 23.
ppi21_in_24[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 24.
ppi21_in_25[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 25.
ppi21_in_26[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 26.
ppi21_in_27[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 27.
ppi21_in_28[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 28.
ppi21_in_29[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 29.
ppi21_in_3[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 3.
ppi21_in_30[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 30.
ppi21_in_31[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 31.
ppi21_in_4[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 4.
ppi21_in_5[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 5.
ppi21_in_6[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 6.

Name	Protocol	Type	Description
ppi21_in_7[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 7.
ppi21_in_8[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 8.
ppi21_in_9[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 9.
ppi22_in_0[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 0.
ppi22_in_1[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 1.
ppi22_in_10[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 10.
ppi22_in_11[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 11.
ppi22_in_12[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 12.
ppi22_in_13[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 13.
ppi22_in_14[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 14.
ppi22_in_15[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 15.
ppi22_in_16[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 16.
ppi22_in_17[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 17.
ppi22_in_18[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 18.
ppi22_in_19[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 19.
ppi22_in_2[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 2.
ppi22_in_20[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 20.
ppi22_in_21[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 21.
ppi22_in_22[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 22.
ppi22_in_23[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 23.
ppi22_in_24[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 24.
ppi22_in_25[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 25.
ppi22_in_26[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 26.
ppi22_in_27[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 27.



Name	Protocol	Type	Description
ppi22_in_28[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 28.
ppi22_in_29[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 29.
ppi22_in_3[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 3.
ppi22_in_30[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 30.
ppi22_in_31[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 31.
ppi22_in_4[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 4.
ppi22_in_5[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 5.
ppi22_in_6[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 6.
ppi22_in_7[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 7.
ppi22_in_8[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 8.
ppi22_in_9[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 9.
ppi23_in_0[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 0.
ppi23_in_1[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 1.
ppi23_in_10[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 10.
ppi23_in_11[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 11.
ppi23_in_12[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 12.
ppi23_in_13[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 13.
ppi23_in_14[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 14.
ppi23_in_15[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 15.
ppi23_in_16[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 16.
ppi23_in_17[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 17.
ppi23_in_18[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 18.
ppi23_in_19[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 19.



Name	Protocol	Type	Description
ppi23_in_2[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 2.
ppi23_in_20[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 20.
ppi23_in_21[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 21.
ppi23_in_22[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 22.
ppi23_in_23[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 23.
ppi23_in_24[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 24.
ppi23_in_25[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 25.
ppi23_in_26[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 26.
ppi23_in_27[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 27.
ppi23_in_28[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 28.
ppi23_in_29[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 29.
ppi23_in_3[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 3.
ppi23_in_30[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 30.
ppi23_in_31[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 31.
ppi23_in_4[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 4.
ppi23_in_5[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 5.
ppi23_in_6[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 6.
ppi23_in_7[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 7.
ppi23_in_8[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 8.
ppi23_in_9[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 9.
ppi24_in_0[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 0.
ppi24_in_1[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 1.
ppi24_in_10[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 10.

Name	Protocol	Type	Description
ppi24_in_11[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 11.
ppi24_in_12[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 12.
ppi24_in_13[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 13.
ppi24_in_14[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 14.
ppi24_in_15[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 15.
ppi24_in_16[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 16.
ppi24_in_17[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 17.
ppi24_in_18[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 18.
ppi24_in_19[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 19.
ppi24_in_2[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 2.
ppi24_in_20[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 20.
ppi24_in_21[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 21.
ppi24_in_22[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 22.
ppi24_in_23[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 23.
ppi24_in_24[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 24.
ppi24_in_25[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 25.
ppi24_in_26[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 26.
ppi24_in_27[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 27.
ppi24_in_28[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 28.
ppi24_in_29[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 29.
ppi24_in_3[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 3.
ppi24_in_30[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 30.
ppi24_in_31[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 31.

Name	Protocol	Type	Description
ppi24_in_4[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 4.
ppi24_in_5[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 5.
ppi24_in_6[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 6.
ppi24_in_7[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 7.
ppi24_in_8[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 8.
ppi24_in_9[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 9.
ppi25_in_0[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 0.
ppi25_in_1[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 1.
ppi25_in_10[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 10.
ppi25_in_11[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 11.
ppi25_in_12[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 12.
ppi25_in_13[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 13.
ppi25_in_14[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 14.
ppi25_in_15[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 15.
ppi25_in_16[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 16.
ppi25_in_17[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 17.
ppi25_in_18[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 18.
ppi25_in_19[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 19.
ppi25_in_2[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 2.
ppi25_in_20[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 20.
ppi25_in_21[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 21.
ppi25_in_22[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 22.
ppi25_in_23[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 23.

Name	Protocol	Type	Description
ppi25_in_24[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 24.
ppi25_in_25[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 25.
ppi25_in_26[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 26.
ppi25_in_27[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 27.
ppi25_in_28[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 28.
ppi25_in_29[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 29.
ppi25_in_3[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 3.
ppi25_in_30[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 30.
ppi25_in_31[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 31.
ppi25_in_4[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 4.
ppi25_in_5[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 5.
ppi25_in_6[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 6.
ppi25_in_7[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 7.
ppi25_in_8[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 8.
ppi25_in_9[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 9.
ppi26_in_0[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 0.
ppi26_in_1[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 1.
ppi26_in_10[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 10.
ppi26_in_11[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 11.
ppi26_in_12[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 12.
ppi26_in_13[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 13.
ppi26_in_14[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 14.
ppi26_in_15[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 15.
ppi26_in_16[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 16.
ppi26_in_17[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 17.
ppi26_in_18[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 18.
ppi26_in_19[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 19.
ppi26_in_2[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 2.
ppi26_in_20[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 20.

Name	Protocol	Type	Description
ppi26_in_21[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 21.
ppi26_in_22[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 22.
ppi26_in_23[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 23.
ppi26_in_24[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 24.
ppi26_in_25[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 25.
ppi26_in_26[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 26.
ppi26_in_27[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 27.
ppi26_in_28[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 28.
ppi26_in_29[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 29.
ppi26_in_3[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 3.
ppi26_in_30[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 30.
ppi26_in_31[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 31.
ppi26_in_4[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 4.
ppi26_in_5[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 5.
ppi26_in_6[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 6.
ppi26_in_7[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 7.
ppi26_in_8[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 8.
ppi26_in_9[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 9.
ppi27_in_0[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 0.
ppi27_in_1[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 1.
ppi27_in_10[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 10.
ppi27_in_11[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 11.
ppi27_in_12[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 12.
ppi27_in_13[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 13.
ppi27_in_14[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 14.
ppi27_in_15[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 15.
ppi27_in_16[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 16.
ppi27_in_17[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 17.
ppi27_in_18[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 18.
ppi27_in_19[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 19.
ppi27_in_2[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 2.
ppi27_in_20[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 20.
ppi27_in_21[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 21.
ppi27_in_22[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 22.
ppi27_in_23[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 23.
ppi27_in_24[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 24.
ppi27_in_25[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 25.
ppi27_in_26[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 26.
ppi27_in_27[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 27.

Name	Protocol	Type	Description
ppi27_in_28[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 28.
ppi27_in_29[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 29.
ppi27_in_3[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 3.
ppi27_in_30[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 30.
ppi27_in_31[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 31.
ppi27_in_4[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 4.
ppi27_in_5[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 5.
ppi27_in_6[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 6.
ppi27_in_7[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 7.
ppi27_in_8[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 8.
ppi27_in_9[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 9.
ppi28_in_0[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 0.
ppi28_in_1[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 1.
ppi28_in_10[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 10.
ppi28_in_11[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 11.
ppi28_in_12[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 12.
ppi28_in_13[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 13.
ppi28_in_14[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 14.
ppi28_in_15[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 15.
ppi28_in_16[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 16.
ppi28_in_17[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 17.
ppi28_in_18[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 18.
ppi28_in_19[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 19.
ppi28_in_2[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 2.
ppi28_in_20[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 20.
ppi28_in_21[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 21.
ppi28_in_22[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 22.
ppi28_in_23[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 23.
ppi28_in_24[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 24.
ppi28_in_25[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 25.
ppi28_in_26[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 26.
ppi28_in_27[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 27.
ppi28_in_28[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 28.
ppi28_in_29[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 29.
ppi28_in_3[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 3.
ppi28_in_30[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 30.
ppi28_in_31[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 31.
ppi28_in_4[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 4.
ppi28_in_5[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 5.

Name	Protocol	Type	Description
ppi28_in_6[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 6.
ppi28_in_7[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 7.
ppi28_in_8[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 8.
ppi28_in_9[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 9.
ppi29_in_0[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 0.
ppi29_in_1[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 1.
ppi29_in_10[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 10.
ppi29_in_11[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 11.
ppi29_in_12[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 12.
ppi29_in_13[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 13.
ppi29_in_14[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 14.
ppi29_in_15[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 15.
ppi29_in_16[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 16.
ppi29_in_17[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 17.
ppi29_in_18[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 18.
ppi29_in_19[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 19.
ppi29_in_2[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 2.
ppi29_in_20[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 20.
ppi29_in_21[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 21.
ppi29_in_22[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 22.
ppi29_in_23[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 23.
ppi29_in_24[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 24.
ppi29_in_25[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 25.
ppi29_in_26[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 26.
ppi29_in_27[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 27.
ppi29_in_28[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 28.
ppi29_in_29[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 29.
ppi29_in_3[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 3.
ppi29_in_30[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 30.
ppi29_in_31[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 31.
ppi29_in_4[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 4.
ppi29_in_5[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 5.
ppi29_in_6[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 6.
ppi29_in_7[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 7.
ppi29_in_8[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 8.
ppi29_in_9[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 9.
ppi30_in_0[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 0.
ppi30_in_1[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 1.



Name	Protocol	Type	Description
ppi30_in_10[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 10.
ppi30_in_11[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 11.
ppi30_in_12[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 12.
ppi30_in_13[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 13.
ppi30_in_14[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 14.
ppi30_in_15[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 15.
ppi30_in_16[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 16.
ppi30_in_17[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 17.
ppi30_in_18[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 18.
ppi30_in_19[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 19.
ppi30_in_2[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 2.
ppi30_in_20[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 20.
ppi30_in_21[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 21.
ppi30_in_22[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 22.
ppi30_in_23[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 23.
ppi30_in_24[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 24.
ppi30_in_25[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 25.
ppi30_in_26[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 26.
ppi30_in_27[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 27.
ppi30_in_28[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 28.
ppi30_in_29[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 29.
ppi30_in_3[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 3.
ppi30_in_30[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 30.



Name	Protocol	Type	Description
ppi30_in_31[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 31.
ppi30_in_4[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 4.
ppi30_in_5[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 5.
ppi30_in_6[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 6.
ppi30_in_7[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 7.
ppi30_in_8[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 8.
ppi30_in_9[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 9.
ppi31_in_0[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 0.
ppi31_in_1[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 1.
ppi31_in_10[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 10.
ppi31_in_11[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 11.
ppi31_in_12[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 12.
ppi31_in_13[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 13.
ppi31_in_14[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 14.
ppi31_in_15[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 15.
ppi31_in_16[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 16.
ppi31_in_17[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 17.
ppi31_in_18[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 18.
ppi31_in_19[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 19.
ppi31_in_2[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 2.
ppi31_in_20[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 20.
ppi31_in_21[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 21.
ppi31_in_22[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 22.
ppi31_in_23[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 23.
ppi31_in_24[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 24.
ppi31_in_25[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 25.
ppi31_in_26[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 26.
ppi31_in_27[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 27.
ppi31_in_28[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 28.
ppi31_in_29[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 29.
ppi31_in_3[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 3.
ppi31_in_30[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 30.
ppi31_in_31[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 31.
ppi31_in_4[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 4.

Name	Protocol	Type	Description
ppi31_in_5[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 5.
ppi31_in_6[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 6.
ppi31_in_7[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 7.
ppi31_in_8[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 8.
ppi31_in_9[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 9.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_0[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 0
redistributor_1[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 1.
redistributor_10[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 10.
redistributor_11[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 11.
redistributor_12[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 12.
redistributor_13[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 13.
redistributor_14[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 14.
redistributor_15[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 15.
redistributor_16[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 16.
redistributor_17[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 17.
redistributor_18[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 18.
redistributor_19[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 19.
redistributor_2[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 2.
redistributor_20[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 20
redistributor_21[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 21.
redistributor_22[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 22.
redistributor_23[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 23
redistributor_24[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 24.
redistributor_25[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 25.
redistributor_26[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 26.
redistributor_27[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 27.
redistributor_28[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 28.
redistributor_29[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 29.
redistributor_3[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 3.
redistributor_30[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 30.
redistributor_31[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 31.
redistributor_4[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 4.
redistributor_5[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 5.
redistributor_6[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 6.
redistributor_7[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 7.
redistributor_8[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 8.
redistributor_9[8]	GICv3Comms	Master	Input from and output to CPU interface for cluster 9.

Name	Protocol	Type	Description
reset	Signal	Slave	Reset.
spi_in[988]	Signal	Slave	Shared peripheral interrupts.
wake_request_0[8]	Signal	Master	Power management outputs of cluster 0.
wake_request_1[8]	Signal	Master	Power management outputs of cluster 1.
wake_request_10[8]	Signal	Master	Power management outputs of cluster 10.
wake_request_11[8]	Signal	Master	Power management outputs of cluster 11.
wake_request_12[8]	Signal	Master	Power management outputs of cluster 12.
wake_request_13[8]	Signal	Master	Power management outputs of cluster 13.
wake_request_14[8]	Signal	Master	Power management outputs of cluster 14.
wake_request_15[8]	Signal	Master	Power management outputs of cluster 15.
wake_request_16[8]	Signal	Master	Power management outputs of cluster 16.
wake_request_17[8]	Signal	Master	Power management outputs of cluster 17.
wake_request_18[8]	Signal	Master	Power management outputs of cluster 18.
wake_request_19[8]	Signal	Master	Power management outputs of cluster 19.
wake_request_2[8]	Signal	Master	Power management outputs of cluster 2.
wake_request_20[8]	Signal	Master	Power management outputs of cluster 20.
wake_request_21[8]	Signal	Master	Power management outputs of cluster 21.
wake_request_22[8]	Signal	Master	Power management outputs of cluster 22.
wake_request_23[8]	Signal	Master	Power management outputs of cluster 23.
wake_request_24[8]	Signal	Master	Power management outputs of cluster 24.
wake_request_25[8]	Signal	Master	Power management outputs of cluster 25.
wake_request_26[8]	Signal	Master	Power management outputs of cluster 26.
wake_request_27[8]	Signal	Master	Power management outputs of cluster 27.
wake_request_28[8]	Signal	Master	Power management outputs of cluster 28.
wake_request_29[8]	Signal	Master	Power management outputs of cluster 29.
wake_request_3[8]	Signal	Master	Power management outputs of cluster 3.
wake_request_30[8]	Signal	Master	Power management outputs of cluster 30.
wake_request_31[8]	Signal	Master	Power management outputs of cluster 31.
wake_request_4[8]	Signal	Master	Power management outputs of cluster 4.
wake_request_5[8]	Signal	Master	Power management outputs of cluster 5.
wake_request_6[8]	Signal	Master	Power management outputs of cluster 6.
wake_request_7[8]	Signal	Master	Power management outputs of cluster 7.
wake_request_8[8]	Signal	Master	Power management outputs of cluster 8.
wake_request_9[8]	Signal	Master	Power management outputs of cluster 9.

## Parameters for GIC500\_ClusterPorts

### GICD\_ITARGETSR-RAZWI

#### Type

bool

**Default value**

0x0

**Description**

If true, the GICD\_ITARGETS registers are RAZ/WI.

**ITS-count****Type**

int

**Default value**

0x1

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**ITS-device-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits supported for ITS device IDs.

**ITS-threaded-command-queue****Type**

bool

**Default value**

0x1

**Description**

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

**SPI-count****Type**

int

**Default value**

0xe0

**Description**

Number of SPIs that are implemented.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**cpus\_per\_cluster\_0****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 0.

**cpus\_per\_cluster\_1****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 1.

**cpus\_per\_cluster\_10****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 10.

**cpus\_per\_cluster\_11****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 11.

**cpus\_per\_cluster\_12****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 12.

**cpus\_per\_cluster\_13****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 13.

**cpus\_per\_cluster\_14****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 14.

**cpus\_per\_cluster\_15****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 15.

**cpus\_per\_cluster\_16****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 16.

**`cpus_per_cluster_17`****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 17.

**`cpus_per_cluster_18`****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 18.

**`cpus_per_cluster_19`****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 19.

**`cpus_per_cluster_2`****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 2.

**`cpus_per_cluster_20`****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 20.

**cpus\_per\_cluster\_21****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 21.

**cpus\_per\_cluster\_22****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 22.

**cpus\_per\_cluster\_23****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 23.

**cpus\_per\_cluster\_24****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 24.

**cpus\_per\_cluster\_25****Type**

int

**Default value**

0x1



**Description**

Number of cores within cluster 25.

**cpus\_per\_cluster\_26****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 26.

**cpus\_per\_cluster\_27****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 27.

**cpus\_per\_cluster\_28****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 28.

**cpus\_per\_cluster\_29****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 29.

**cpus\_per\_cluster\_3****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 3.

**cpus\_per\_cluster\_30****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 30.

**cpus\_per\_cluster\_31****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 31.

**cpus\_per\_cluster\_4****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 4.

**cpus\_per\_cluster\_5****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 5.

**cpus\_per\_cluster\_6****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 6.

**cpus\_per\_cluster\_7****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 7.

**cpus\_per\_cluster\_8****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 8.

**cpus\_per\_cluster\_9****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 9.

**delay-ITS-accesses****Type**

bool

**Default value**

0x1

**Description**

Delay accesses from the ITS until GICR\_SYNCRR is read.

**delay-redistributor-accesses****Type**

bool

**Default value**

0x1

**Description**

Delay memory accesses from the redistributor until GICR\_SYNCR is read.

**enable\_protocol\_checking****Type**

bool

**Default value**

0x0

**Description**

Enable/disable protocol checking at cpu interface.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**has-two-security-states****Type**

bool

**Default value**

0x1

**Description**

If true, has two security states.

**num\_clusters****Type**

int

**Default value**

0x1

**Description**

Number of implemented affinity level1 clusters.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**redistributor-threaded-sync****Type**

bool

**Default value**

0x1

**Description**

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

**reg-base****Type**

int

**Default value**

0x10000000

**Description**

GIC500 base address.

**using-generated-memorymap****Type**

bool

**Default value**

0x1

**Description**

Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.

**wakeup-on-reset****Type**

bool

**Default value**

0x0

**Description**

Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

### 3.10.33 GIC500\_Filter

GIC500 Component for distribution of interrupts, filtering version for validation. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-969: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About GIC500\_Filter

This is a single-component implementation of the GICv3 architecture with support for 256 cores. You can configure the model to support a maximum of 32 clusters with 8 cores per cluster. Use it with an Armv8-A core to deliver interrupts. It supports a single Interrupt Translation Service for message-based interrupts. It supports the architectural features, but does not support the implementation defined features.



Note

- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- For writes to `GITS_TRANSLATE64R`, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see [MSIRewriter](#).

#### MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-970: MasterID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



Note

The `pvbus_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

#### Iris and MTI instances for GIC500\_Filter

This model has the following Iris instances:

**Table 3-971: GIC500\_Filter Iris instances**

InstanceName	ComponentName
GIC500_Filter	GIC_IRI
GIC500_Filter.ITS0	GICv3InterruptTranslationService
GIC500_Filter.rd_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0_0_0	GICv3Redistributor
GIC500_Filter.rd_tl	GICv3Distributor

This model has the following MTI trace components:

**Table 3-972: GIC500\_Filter MTI instances**

InstanceName	ComponentName
GIC500_Filter.ITS0	GICv3InterruptTranslationService
GIC500_Filter.rd_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0_0_0	GICv3Redistributor
GIC500_Filter.rd_tl	GICv3Distributor

### Ports for GIC500\_Filter

**Table 3-973: Ports**

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Disable some SPIs signal.
cpu_active_0[8]	Signal	Slave	cpu_active pins of cluster 0.
cpu_active_1[8]	Signal	Slave	cpu_active pins of cluster 1.
cpu_active_10[8]	Signal	Slave	cpu_active pins of cluster 10.
cpu_active_11[8]	Signal	Slave	cpu_active pins of cluster 11.
cpu_active_12[8]	Signal	Slave	cpu_active pins of cluster 12.
cpu_active_13[8]	Signal	Slave	cpu_active pins of cluster 13.
cpu_active_14[8]	Signal	Slave	cpu_active pins of cluster 14.
cpu_active_15[8]	Signal	Slave	cpu_active pins of cluster 15.
cpu_active_16[8]	Signal	Slave	cpu_active pins of cluster 16.
cpu_active_17[8]	Signal	Slave	cpu_active pins of cluster 17.
cpu_active_18[8]	Signal	Slave	cpu_active pins of cluster 18.
cpu_active_19[8]	Signal	Slave	cpu_active pins of cluster 19.
cpu_active_2[8]	Signal	Slave	cpu_active pins of cluster 2.
cpu_active_20[8]	Signal	Slave	cpu_active pins of cluster 20.
cpu_active_21[8]	Signal	Slave	cpu_active pins of cluster 21.
cpu_active_22[8]	Signal	Slave	cpu_active pins of cluster 22.

Name	Protocol	Type	Description
cpu_active_23[8]	Signal	Slave	cpu_active pins of cluster 23.
cpu_active_24[8]	Signal	Slave	cpu_active pins of cluster 24.
cpu_active_25[8]	Signal	Slave	cpu_active pins of cluster 25.
cpu_active_26[8]	Signal	Slave	cpu_active pins of cluster 26.
cpu_active_27[8]	Signal	Slave	cpu_active pins of cluster 27.
cpu_active_28[8]	Signal	Slave	cpu_active pins of cluster 28.
cpu_active_29[8]	Signal	Slave	cpu_active pins of cluster 29.
cpu_active_3[8]	Signal	Slave	cpu_active pins of cluster 3.
cpu_active_30[8]	Signal	Slave	cpu_active pins of cluster 30.
cpu_active_31[8]	Signal	Slave	cpu_active pins of cluster 31.
cpu_active_4[8]	Signal	Slave	cpu_active pins of cluster 4.
cpu_active_5[8]	Signal	Slave	cpu_active pins of cluster 5.
cpu_active_6[8]	Signal	Slave	cpu_active pins of cluster 6.
cpu_active_7[8]	Signal	Slave	cpu_active pins of cluster 7.
cpu_active_8[8]	Signal	Slave	cpu_active pins of cluster 8.
cpu_active_9[8]	Signal	Slave	cpu_active pins of cluster 9.
po_reset	Signal	Slave	Power on reset.
ppi16_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 0.
ppi16_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 1.
ppi16_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 10.
ppi16_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 11.
ppi16_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 12.
ppi16_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 13.
ppi16_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 14.
ppi16_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 15.
ppi16_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 16.
ppi16_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 17.
ppi16_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 18.
ppi16_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 19.
ppi16_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 2.
ppi16_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 20.
ppi16_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 21.
ppi16_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 22.
ppi16_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 23.
ppi16_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 24.
ppi16_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 25.
ppi16_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 26.
ppi16_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 27.
ppi16_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 28.



Name	Protocol	Type	Description
ppi16_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 29.
ppi16_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 3.
ppi16_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 30.
ppi16_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 31.
ppi16_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 4.
ppi16_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 5.
ppi16_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 6.
ppi16_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 7.
ppi16_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 8.
ppi16_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID16) of cluster 9.
ppi17_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 0.
ppi17_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 1.
ppi17_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 10.
ppi17_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 11.
ppi17_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 12.
ppi17_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 13.
ppi17_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 14.
ppi17_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 15.
ppi17_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 16.
ppi17_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 17.
ppi17_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 18.
ppi17_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 19.
ppi17_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 2.
ppi17_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 20.
ppi17_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 21.
ppi17_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 22.
ppi17_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 23.
ppi17_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 24.
ppi17_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 25.
ppi17_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 26.
ppi17_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 27.
ppi17_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 28.
ppi17_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 29.
ppi17_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 3.
ppi17_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 30.
ppi17_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 31.
ppi17_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 4.
ppi17_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 5.
ppi17_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 6.

Name	Protocol	Type	Description
ppi17_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 7.
ppi17_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 8.
ppi17_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID17) of cluster 9.
ppi18_in_0[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 0.
ppi18_in_1[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 1.
ppi18_in_10[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 10.
ppi18_in_11[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 11.
ppi18_in_12[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 12.
ppi18_in_13[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 13.
ppi18_in_14[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 14.
ppi18_in_15[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 15.
ppi18_in_16[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 16.
ppi18_in_17[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 17.
ppi18_in_18[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 18.
ppi18_in_19[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 19.
ppi18_in_2[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 2.
ppi18_in_20[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 20.
ppi18_in_21[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 21.
ppi18_in_22[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 22.
ppi18_in_23[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 23.
ppi18_in_24[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 24.
ppi18_in_25[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 25.
ppi18_in_26[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 26.
ppi18_in_27[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 27.
ppi18_in_28[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 28.
ppi18_in_29[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 29.
ppi18_in_3[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 3.
ppi18_in_30[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 30.
ppi18_in_31[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 31.
ppi18_in_4[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 4.
ppi18_in_5[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 5.
ppi18_in_6[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 6.
ppi18_in_7[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 7.
ppi18_in_8[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 8.
ppi18_in_9[8]	Signal	Slave	Private peripheral interrupts interrupts (ID18) of cluster 9.
ppi19_in_0[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 0.
ppi19_in_1[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 1.
ppi19_in_10[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 10.
ppi19_in_11[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 11.

Name	Protocol	Type	Description
ppi19_in_12[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 12.
ppi19_in_13[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 13.
ppi19_in_14[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 14.
ppi19_in_15[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 15.
ppi19_in_16[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 16.
ppi19_in_17[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 17.
ppi19_in_18[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 18.
ppi19_in_19[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 19.
ppi19_in_2[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 2.
ppi19_in_20[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 20.
ppi19_in_21[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 21.
ppi19_in_22[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 22.
ppi19_in_23[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 23.
ppi19_in_24[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 24.
ppi19_in_25[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 25.
ppi19_in_26[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 26.
ppi19_in_27[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 27.
ppi19_in_28[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 28.
ppi19_in_29[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 29.
ppi19_in_3[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 3.
ppi19_in_30[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 30.
ppi19_in_31[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 31.
ppi19_in_4[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 4.
ppi19_in_5[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 5.
ppi19_in_6[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 6.
ppi19_in_7[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 7.
ppi19_in_8[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 8.
ppi19_in_9[8]	Signal	Slave	Private peripheral interrupts (ID19) of cluster 9.
ppi20_in_0[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 0.
ppi20_in_1[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 1.
ppi20_in_10[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 10.
ppi20_in_11[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 11.
ppi20_in_12[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 12.
ppi20_in_13[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 13.
ppi20_in_14[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 14.
ppi20_in_15[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 15.
ppi20_in_16[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 16.
ppi20_in_17[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 17.
ppi20_in_18[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 18.

Name	Protocol	Type	Description
ppi20_in_19[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 19.
ppi20_in_2[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 2.
ppi20_in_20[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 20.
ppi20_in_21[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 21.
ppi20_in_22[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 22.
ppi20_in_23[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 23.
ppi20_in_24[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 24.
ppi20_in_25[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 25.
ppi20_in_26[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 26.
ppi20_in_27[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 27.
ppi20_in_28[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 28.
ppi20_in_29[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 29.
ppi20_in_3[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 3.
ppi20_in_30[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 30.
ppi20_in_31[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 31.
ppi20_in_4[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 4.
ppi20_in_5[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 5.
ppi20_in_6[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 6.
ppi20_in_7[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 7.
ppi20_in_8[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 8.
ppi20_in_9[8]	Signal	Slave	Private peripheral interrupts (ID20) of cluster 9.
ppi21_in_0[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 0.
ppi21_in_1[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 1.
ppi21_in_10[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 10.
ppi21_in_11[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 11.
ppi21_in_12[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 12.
ppi21_in_13[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 13.
ppi21_in_14[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 14.
ppi21_in_15[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 15.
ppi21_in_16[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 16.
ppi21_in_17[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 17.
ppi21_in_18[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 18.
ppi21_in_19[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 19.
ppi21_in_2[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 2.
ppi21_in_20[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 20.
ppi21_in_21[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 21.
ppi21_in_22[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 22.
ppi21_in_23[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 23.
ppi21_in_24[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 24.

Name	Protocol	Type	Description
ppi21_in_25[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 25.
ppi21_in_26[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 26.
ppi21_in_27[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 27.
ppi21_in_28[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 28.
ppi21_in_29[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 29.
ppi21_in_3[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 3.
ppi21_in_30[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 30.
ppi21_in_31[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 31.
ppi21_in_4[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 4.
ppi21_in_5[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 5.
ppi21_in_6[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 6.
ppi21_in_7[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 7.
ppi21_in_8[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 8.
ppi21_in_9[8]	Signal	Slave	Private peripheral interrupts (ID21) of cluster 9.
ppi22_in_0[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 0.
ppi22_in_1[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 1.
ppi22_in_10[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 10.
ppi22_in_11[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 11.
ppi22_in_12[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 12.
ppi22_in_13[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 13.
ppi22_in_14[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 14.
ppi22_in_15[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 15.
ppi22_in_16[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 16.
ppi22_in_17[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 17.
ppi22_in_18[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 18.
ppi22_in_19[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 19.
ppi22_in_2[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 2.
ppi22_in_20[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 20.
ppi22_in_21[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 21.

Name	Protocol	Type	Description
ppi22_in_22[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 22.
ppi22_in_23[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 23.
ppi22_in_24[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 24.
ppi22_in_25[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 25.
ppi22_in_26[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 26.
ppi22_in_27[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 27.
ppi22_in_28[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 28.
ppi22_in_29[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 29.
ppi22_in_3[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 3.
ppi22_in_30[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 30.
ppi22_in_31[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 31.
ppi22_in_4[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 4.
ppi22_in_5[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 5.
ppi22_in_6[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 6.
ppi22_in_7[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 7.
ppi22_in_8[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 8.
ppi22_in_9[8]	Signal	Slave	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 9.
ppi23_in_0[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 0.
ppi23_in_1[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 1.
ppi23_in_10[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 10.
ppi23_in_11[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 11.
ppi23_in_12[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 12.
ppi23_in_13[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 13.

Name	Protocol	Type	Description
ppi23_in_14[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 14.
ppi23_in_15[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 15.
ppi23_in_16[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 16.
ppi23_in_17[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 17.
ppi23_in_18[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 18.
ppi23_in_19[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 19.
ppi23_in_2[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 2.
ppi23_in_20[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 20.
ppi23_in_21[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 21.
ppi23_in_22[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 22.
ppi23_in_23[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 23.
ppi23_in_24[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 24.
ppi23_in_25[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 25.
ppi23_in_26[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 26.
ppi23_in_27[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 27.
ppi23_in_28[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 28.
ppi23_in_29[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 29.
ppi23_in_3[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 3.
ppi23_in_30[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 30.
ppi23_in_31[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 31.
ppi23_in_4[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 4.
ppi23_in_5[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 5.
ppi23_in_6[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 6.



Name	Protocol	Type	Description
ppi23_in_7[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 7.
ppi23_in_8[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 8.
ppi23_in_9[8]	Signal	Slave	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 9.
ppi24_in_0[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 0.
ppi24_in_1[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 1.
ppi24_in_10[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 10.
ppi24_in_11[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 11.
ppi24_in_12[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 12.
ppi24_in_13[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 13.
ppi24_in_14[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 14.
ppi24_in_15[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 15.
ppi24_in_16[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 16.
ppi24_in_17[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 17.
ppi24_in_18[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 18.
ppi24_in_19[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 19.
ppi24_in_2[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 2.
ppi24_in_20[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 20.
ppi24_in_21[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 21.
ppi24_in_22[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 22.
ppi24_in_23[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 23.
ppi24_in_24[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 24.
ppi24_in_25[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 25.
ppi24_in_26[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 26.



Name	Protocol	Type	Description
ppi24_in_27[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 27.
ppi24_in_28[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 28.
ppi24_in_29[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 29.
ppi24_in_3[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 3.
ppi24_in_30[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 30.
ppi24_in_31[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 31.
ppi24_in_4[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 4.
ppi24_in_5[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 5.
ppi24_in_6[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 6.
ppi24_in_7[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 7.
ppi24_in_8[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 8.
ppi24_in_9[8]	Signal	Slave	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 9.
ppi25_in_0[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 0.
ppi25_in_1[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 1.
ppi25_in_10[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 10.
ppi25_in_11[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 11.
ppi25_in_12[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 12.
ppi25_in_13[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 13.
ppi25_in_14[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 14.
ppi25_in_15[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 15.
ppi25_in_16[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 16.
ppi25_in_17[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 17.
ppi25_in_18[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 18.

Name	Protocol	Type	Description
ppi25_in_19[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 19.
ppi25_in_2[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 2.
ppi25_in_20[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 20.
ppi25_in_21[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 21.
ppi25_in_22[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 22.
ppi25_in_23[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 23.
ppi25_in_24[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 24.
ppi25_in_25[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 25.
ppi25_in_26[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 26.
ppi25_in_27[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 27.
ppi25_in_28[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 28.
ppi25_in_29[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 29.
ppi25_in_3[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 3.
ppi25_in_30[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 30.
ppi25_in_31[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 31.
ppi25_in_4[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 4.
ppi25_in_5[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 5.
ppi25_in_6[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 6.
ppi25_in_7[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 7.
ppi25_in_8[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 8.
ppi25_in_9[8]	Signal	Slave	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 9.
ppi26_in_0[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 0.
ppi26_in_1[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 1.
ppi26_in_10[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 10.
ppi26_in_11[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 11.

Name	Protocol	Type	Description
ppi26_in_12[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 12.
ppi26_in_13[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 13.
ppi26_in_14[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 14.
ppi26_in_15[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 15.
ppi26_in_16[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 16.
ppi26_in_17[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 17.
ppi26_in_18[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 18.
ppi26_in_19[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 19.
ppi26_in_2[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 2.
ppi26_in_20[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 20.
ppi26_in_21[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 21.
ppi26_in_22[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 22.
ppi26_in_23[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 23.
ppi26_in_24[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 24.
ppi26_in_25[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 25.
ppi26_in_26[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 26.
ppi26_in_27[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 27.
ppi26_in_28[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 28.
ppi26_in_29[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 29.
ppi26_in_3[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 3.
ppi26_in_30[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 30.
ppi26_in_31[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 31.
ppi26_in_4[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 4.
ppi26_in_5[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 5.
ppi26_in_6[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 6.
ppi26_in_7[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 7.
ppi26_in_8[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 8.
ppi26_in_9[8]	Signal	Slave	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 9.
ppi27_in_0[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 0.
ppi27_in_1[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 1.
ppi27_in_10[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 10.
ppi27_in_11[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 11.
ppi27_in_12[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 12.
ppi27_in_13[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 13.
ppi27_in_14[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 14.
ppi27_in_15[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 15.
ppi27_in_16[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 16.
ppi27_in_17[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 17.
ppi27_in_18[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 18.

Name	Protocol	Type	Description
ppi27_in_19[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 19.
ppi27_in_2[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 2.
ppi27_in_20[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 20.
ppi27_in_21[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 21.
ppi27_in_22[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 22.
ppi27_in_23[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 23.
ppi27_in_24[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 24.
ppi27_in_25[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 25.
ppi27_in_26[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 26.
ppi27_in_27[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 27.
ppi27_in_28[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 28.
ppi27_in_29[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 29.
ppi27_in_3[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 3.
ppi27_in_30[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 30.
ppi27_in_31[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 31.
ppi27_in_4[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 4.
ppi27_in_5[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 5.
ppi27_in_6[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 6.
ppi27_in_7[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 7.
ppi27_in_8[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 8.
ppi27_in_9[8]	Signal	Slave	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 9.
ppi28_in_0[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 0.
ppi28_in_1[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 1.
ppi28_in_10[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 10.
ppi28_in_11[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 11.
ppi28_in_12[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 12.
ppi28_in_13[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 13.
ppi28_in_14[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 14.
ppi28_in_15[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 15.
ppi28_in_16[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 16.
ppi28_in_17[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 17.
ppi28_in_18[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 18.
ppi28_in_19[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 19.
ppi28_in_2[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 2.
ppi28_in_20[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 20.
ppi28_in_21[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 21.
ppi28_in_22[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 22.
ppi28_in_23[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 23.
ppi28_in_24[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 24.

Name	Protocol	Type	Description
ppi28_in_25[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 25.
ppi28_in_26[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 26.
ppi28_in_27[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 27.
ppi28_in_28[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 28.
ppi28_in_29[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 29.
ppi28_in_3[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 3.
ppi28_in_30[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 30.
ppi28_in_31[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 31.
ppi28_in_4[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 4.
ppi28_in_5[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 5.
ppi28_in_6[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 6.
ppi28_in_7[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 7.
ppi28_in_8[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 8.
ppi28_in_9[8]	Signal	Slave	Private peripheral interrupts (ID28) of cluster 9.
ppi29_in_0[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 0.
ppi29_in_1[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 1.
ppi29_in_10[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 10.
ppi29_in_11[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 11.
ppi29_in_12[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 12.
ppi29_in_13[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 13.
ppi29_in_14[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 14.
ppi29_in_15[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 15.
ppi29_in_16[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 16.
ppi29_in_17[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 17.
ppi29_in_18[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 18.
ppi29_in_19[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 19.
ppi29_in_2[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 2.
ppi29_in_20[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 20.
ppi29_in_21[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 21.
ppi29_in_22[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 22.
ppi29_in_23[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 23.
ppi29_in_24[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 24.
ppi29_in_25[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 25.
ppi29_in_26[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 26.
ppi29_in_27[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 27.
ppi29_in_28[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 28.
ppi29_in_29[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 29.
ppi29_in_3[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 3.
ppi29_in_30[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 30.

Name	Protocol	Type	Description
ppi29_in_31[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 31.
ppi29_in_4[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 4.
ppi29_in_5[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 5.
ppi29_in_6[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 6.
ppi29_in_7[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 7.
ppi29_in_8[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 8.
ppi29_in_9[8]	Signal	Slave	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 9.
ppi30_in_0[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 0.
ppi30_in_1[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 1.
ppi30_in_10[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 10.
ppi30_in_11[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 11.
ppi30_in_12[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 12.
ppi30_in_13[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 13.
ppi30_in_14[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 14.
ppi30_in_15[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 15.
ppi30_in_16[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 16.
ppi30_in_17[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 17.
ppi30_in_18[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 18.
ppi30_in_19[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 19.
ppi30_in_2[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 2.
ppi30_in_20[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 20.
ppi30_in_21[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 21.
ppi30_in_22[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 22.
ppi30_in_23[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 23.
ppi30_in_24[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 24.
ppi30_in_25[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 25.

Name	Protocol	Type	Description
ppi30_in_26[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 26.
ppi30_in_27[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 27.
ppi30_in_28[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 28.
ppi30_in_29[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 29.
ppi30_in_3[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 3.
ppi30_in_30[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 30.
ppi30_in_31[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 31.
ppi30_in_4[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 4.
ppi30_in_5[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 5.
ppi30_in_6[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 6.
ppi30_in_7[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 7.
ppi30_in_8[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 8.
ppi30_in_9[8]	Signal	Slave	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 9.
ppi31_in_0[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 0.
ppi31_in_1[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 1.
ppi31_in_10[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 10.
ppi31_in_11[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 11.
ppi31_in_12[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 12.
ppi31_in_13[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 13.
ppi31_in_14[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 14.
ppi31_in_15[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 15.
ppi31_in_16[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 16.
ppi31_in_17[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 17.
ppi31_in_18[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 18.
ppi31_in_19[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 19.
ppi31_in_2[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 2.
ppi31_in_20[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 20.
ppi31_in_21[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 21.
ppi31_in_22[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 22.
ppi31_in_23[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 23.



Name	Protocol	Type	Description
ppi31_in_24[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 24.
ppi31_in_25[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 25.
ppi31_in_26[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 26.
ppi31_in_27[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 27.
ppi31_in_28[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 28.
ppi31_in_29[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 29.
ppi31_in_3[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 3.
ppi31_in_30[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 30.
ppi31_in_31[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 31.
ppi31_in_4[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 4.
ppi31_in_5[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 5.
ppi31_in_6[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 6.
ppi31_in_7[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 7.
ppi31_in_8[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 8.
ppi31_in_9[8]	Signal	Slave	Private peripheral interrupts (ID31) of cluster 9.
pvbus_filtermiss_m	PVBus	Master	passthrough for transactions not targetting one of the pages associated with the IRI.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Reset.
spi_in[988]	Signal	Slave	Shared peripheral interrupts.
wake_request_0[8]	Signal	Master	Power management outputs of cluster 0.
wake_request_1[8]	Signal	Master	Power management outputs of cluster 1.
wake_request_10[8]	Signal	Master	Power management outputs of cluster 10.
wake_request_11[8]	Signal	Master	Power management outputs of cluster 11.
wake_request_12[8]	Signal	Master	Power management outputs of cluster 12.
wake_request_13[8]	Signal	Master	Power management outputs of cluster 13.
wake_request_14[8]	Signal	Master	Power management outputs of cluster 14.
wake_request_15[8]	Signal	Master	Power management outputs of cluster 15.
wake_request_16[8]	Signal	Master	Power management outputs of cluster 16.
wake_request_17[8]	Signal	Master	Power management outputs of cluster 17.
wake_request_18[8]	Signal	Master	Power management outputs of cluster 18.
wake_request_19[8]	Signal	Master	Power management outputs of cluster 19.
wake_request_2[8]	Signal	Master	Power management outputs of cluster 2.
wake_request_20[8]	Signal	Master	Power management outputs of cluster 20.
wake_request_21[8]	Signal	Master	Power management outputs of cluster 21.
wake_request_22[8]	Signal	Master	Power management outputs of cluster 22.
wake_request_23[8]	Signal	Master	Power management outputs of cluster 23.
wake_request_24[8]	Signal	Master	Power management outputs of cluster 24.



Name	Protocol	Type	Description
wake_request_25[8]	Signal	Master	Power management outputs of cluster 25.
wake_request_26[8]	Signal	Master	Power management outputs of cluster 26.
wake_request_27[8]	Signal	Master	Power management outputs of cluster 27.
wake_request_28[8]	Signal	Master	Power management outputs of cluster 28.
wake_request_29[8]	Signal	Master	Power management outputs of cluster 29.
wake_request_3[8]	Signal	Master	Power management outputs of cluster 3.
wake_request_30[8]	Signal	Master	Power management outputs of cluster 30.
wake_request_31[8]	Signal	Master	Power management outputs of cluster 31.
wake_request_4[8]	Signal	Master	Power management outputs of cluster 4.
wake_request_5[8]	Signal	Master	Power management outputs of cluster 5.
wake_request_6[8]	Signal	Master	Power management outputs of cluster 6.
wake_request_7[8]	Signal	Master	Power management outputs of cluster 7.
wake_request_8[8]	Signal	Master	Power management outputs of cluster 8.
wake_request_9[8]	Signal	Master	Power management outputs of cluster 9.

## Parameters for GIC500\_Filter

### **GICD\_ITARGETSR-RAZWI**

#### Type

bool

#### Default value

0x0

#### Description

If true, the GICD\_ITARGETS registers are RAZ/WI.

### **ITS-count**

#### Type

int

#### Default value

0x1

#### Description

Number of Interrupt Translation Services to be instantiated (0=none).

### **ITS-device-bits**

#### Type

int

#### Default value

0x10

**Description**

Number of bits supported for ITS device IDs.

**ITS-threaded-command-queue****Type**

bool

**Default value**

0x1

**Description**

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

**SPI-count****Type**

int

**Default value**

0xe0

**Description**

Number of SPIs that are implemented.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**cpus\_per\_cluster\_0****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 0.

**cpus\_per\_cluster\_1****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 1.

**cpus\_per\_cluster\_10****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 10.

**cpus\_per\_cluster\_11****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 11.

**cpus\_per\_cluster\_12****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 12.

**cpus\_per\_cluster\_13****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 13.

**cpus\_per\_cluster\_14****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 14.

**cpus\_per\_cluster\_15****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 15.

**cpus\_per\_cluster\_16****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 16.

**cpus\_per\_cluster\_17****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 17.

**cpus\_per\_cluster\_18****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 18.

**cpus\_per\_cluster\_19****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 19.

**cpus\_per\_cluster\_2****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 2.

**cpus\_per\_cluster\_20****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 20.

**cpus\_per\_cluster\_21****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 21.

**cpus\_per\_cluster\_22****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 22.

**cpus\_per\_cluster\_23****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 23.

**cpus\_per\_cluster\_24****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 24.

**cpus\_per\_cluster\_25****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 25.

**cpus\_per\_cluster\_26****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 26.

**cpus\_per\_cluster\_27****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 27.

**cpus\_per\_cluster\_28****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 28.

**cpus\_per\_cluster\_29****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 29.

**cpus\_per\_cluster\_3****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 3.

**cpus\_per\_cluster\_30****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 30.

**cpus\_per\_cluster\_31****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 31.

**cpus\_per\_cluster\_4****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 4.

**cpus\_per\_cluster\_5****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 5.

**cpus\_per\_cluster\_6****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 6.

**cpus\_per\_cluster\_7****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 7.

**cpus\_per\_cluster\_8****Type**

int

**Default value**

0x1

**Description**

Number of cores within cluster 8.

**cpus\_per\_cluster\_9****Type**

int



**Default value**

0x1

**Description**

Number of cores within cluster 9.

**delay-ITS-accesses****Type**

bool

**Default value**

0x1

**Description**

Delay accesses from the ITS until GICR\_SYNCNR is read.

**delay-redistributor-accesses****Type**

bool

**Default value**

0x1

**Description**

Delay memory accesses from the redistributor until GICR\_SYNCNR is read.

**enable\_protocol\_checking****Type**

bool

**Default value**

0x0

**Description**

Enable/disable protocol checking at cpu interface.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**has-two-security-states****Type**

bool

**Default value**

0x1

**Description**

If true, has two security states.

**num\_clusters****Type**

int

**Default value**

0x1

**Description**

Number of implemented affinity level1 clusters.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**redistributor-threaded-sync****Type**

bool

**Default value**

0x1

**Description**

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

**reg-base****Type**

int

**Default value**

0x10000000

**Description**

GIC500 base address.

**using-generated-memorymap**

**Type**  
bool

**Default value**  
0x1

**Description**  
Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.

**wakeup-on-reset**

**Type**  
bool

**Default value**  
0x0

**Description**  
Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

3.10.34 GIC600

GIC-600 IRI implementation: Single chip validation/filter component variant limited to 265 PE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-974: IP revisions support

Revision	Quality level
r1p6	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About GIC600

GIC600 and GIC600\_Filter are minimal models of an Arm GIC-600 Generic Interrupt Controller, suitable for single-chip or multichip systems, where multichip operation has been tested, with the following exceptions:

- LPI/LPI command is implemented but untested.
- Power operation is unimplemented.

They provide a simple configuration interface that allows designers to introduce GIC600-like functionality to their systems, while only implementing the architectural behavior, as defined by the GICv3 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC600 and GIC600\_Filter have a `pvbus_s` port for register accesses and a `pvbus_m` port for the LPI-related traffic from redistributors and the ITS. The `pvbus_m` port is also used to compose multichip operation.

In addition, the GIC600\_Filter variant has a `pvbus_filtermiss_m` port, to which any transactions on the `pvbus_s` port and not directed to a 4K page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC600 variant.

It is recommended to use the GIC600 variant in most cases.

Also, the RAS register is implemented and various RAS errors are reported through status registers. Not all the RAS errors are available because Fast Models does not model errors that can cause ECC errors. Fast Models supports error record 0 and error record 13+.



Note

- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- For writes to `GITS_TRANSLATE64R`, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see [MSIRewriter](#).

## MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-975: MasterID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



Note

The `pvbus_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

## Iris and MTI instances for GIC600

This model has the following Iris instances:

**Table 3-976: GIC600 Iris instances**

InstanceName	ComponentName
GIC600	GIC_IRI
GIC600.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600.ITS0	GICv3InterruptTranslationService
GIC600.rd_0	GICv3RedistributorInternal
GIC600.rd_0_0	GICv3RedistributorInternal
GIC600.rd_0_0_0	GICv3RedistributorInternal
GIC600.rd_0_0_0_0	GICv3Redistributor
GIC600.rd_t1	GICv3Distributor

This model has the following MTI trace components:

**Table 3-977: GIC600 MTI instances**

InstanceName	ComponentName
GIC600.GICV3_ProtocolChecker	<a href="#">GICv3ProtocolChecker</a>
GIC600.ITS0	<a href="#">GICv3InterruptTranslationService</a>
GIC600.rd_0	<a href="#">GICv3RedistributorInternal</a>
GIC600.rd_0_0	<a href="#">GICv3RedistributorInternal</a>
GIC600.rd_0_0_0	<a href="#">GICv3RedistributorInternal</a>
GIC600.rd_0_0_0_0	<a href="#">GICv3Redistributor</a>
GIC600.rd_t1	<a href="#">GICv3Distributor</a>

## Ports for GIC600

**Table 3-978: Ports**

Name	Protocol	Type	Description
chip_id	<a href="#">Value</a>	Slave	chip_id port which is valid from GIC600 r1p2. Writing to this port for prior GIC600 version will be ignored.
cpu_active_s[256]	<a href="#">Signal</a>	Slave	CPUActive pins.
po_reset	<a href="#">Signal</a>	Slave	Resets.
ppi_in_0[16]	<a href="#">Signal</a>	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1[16]	<a href="#">Signal</a>	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_10[16]	<a href="#">Signal</a>	Slave	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_100[16]	<a href="#">Signal</a>	Slave	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101[16]	<a href="#">Signal</a>	Slave	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102[16]	<a href="#">Signal</a>	Slave	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103[16]	<a href="#">Signal</a>	Slave	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104[16]	<a href="#">Signal</a>	Slave	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105[16]	<a href="#">Signal</a>	Slave	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106[16]	<a href="#">Signal</a>	Slave	Private peripheral interrupts (ID16-ID31) for cpu 106.
ppi_in_107[16]	<a href="#">Signal</a>	Slave	Private peripheral interrupts (ID16-ID31) for cpu 107.

Name	Protocol	Type	Description
ppi_in_108[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 109.
ppi_in_11[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_110[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 113.
ppi_in_114[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 114.
ppi_in_115[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 118.
ppi_in_119[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_12[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_120[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 120.
ppi_in_121[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 122.
ppi_in_123[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 123.
ppi_in_124[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 126.
ppi_in_127[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 128.
ppi_in_129[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_13[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_130[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_14[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_140[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 141.
ppi_in_142[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 142.

Name	Protocol	Type	Description
ppi_in_143[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 145.
ppi_in_146[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 149.
ppi_in_15[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_150[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 150.
ppi_in_151[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 153.
ppi_in_154[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 157.
ppi_in_158[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 159.
ppi_in_16[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_160[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 162.
ppi_in_163[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 164.
ppi_in_165[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_17[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_170[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 177.
ppi_in_178[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 178.

Name	Protocol	Type	Description
ppi_in_179[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_18[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_180[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 181.
ppi_in_182[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 184.
ppi_in_185[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 187.
ppi_in_188[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 189.
ppi_in_19[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_190[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 193.
ppi_in_194[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 194.
ppi_in_195[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 195.
ppi_in_196[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 199.
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_20[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_200[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_21[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_210[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 211.
ppi_in_212[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 212.



Name	Protocol	Type	Description
ppi_in_213[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 217.
ppi_in_218[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 219.
ppi_in_22[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_220[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 222.
ppi_in_223[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 223.
ppi_in_224[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 229.
ppi_in_23[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_230[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 230.
ppi_in_231[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 234.
ppi_in_235[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 236.
ppi_in_237[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_24[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_240[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 247.
ppi_in_248[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 248.

Name	Protocol	Type	Description
ppi_in_249[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_25[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_250[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 253.
ppi_in_254[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 254.
ppi_in_255[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_26[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_30[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_40[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_50[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 53.

Name	Protocol	Type	Description
ppi_in_54[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_60[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_70[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_8[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_80[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 89.

Name	Protocol	Type	Description
ppi_in_9[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 9.
ppi_in_90[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 95.
ppi_in_96[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 99.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Resets.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.

## Parameters for GIC600

### CPU-affinities

#### Type

string

#### Default value

0.0.0.0

#### Description

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

### CPU-affinities-file

#### Type

string

#### Default value

""

#### Description

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

**DS-behaviour****Type**

int

**Default value**

2

GICD\_CTLR.DS field behaviour

0 :RAZ/WI

1 :RAO/WI

2 :RW

**IIDR****Type**

int

**Default value**

0x201743b

**Description**

GICD\_IIDR and GICR\_IIDR value. Defaults to the latest revision.

**ITS-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of interrupt bits supported by ITS.

**ITS-collection-ID-bits****Type**

int

**Default value**

0x8

**Description**

Number of collection bits supported by ITS (optional parameter, 0 =&gt; 16bits support and GITS\_TYPER.CIL=0).

**ITS-count****Type**

int

**Default value**

0x1

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**ITS-device-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits supported for ITS device IDs.

**PPI-count****Type**

int

**Default value**

16

Selects the number of PPI available for each PE:

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31

**RAS-CFI-support****Type**

bool

**Default value**

0x0

**Description**

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

**RAS-FI-support****Type**

bool

**Default value**

0x0

**Description**

If true, fault handling interrupt is supported. Not supported otherwise.

**RAS-UE-support****Type**

bool

**Default value**

0x0

**Description**

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

**RAS-UI-support****Type**

bool

**Default value**

0x0

**Description**

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

**SPI-blocks****Type**

int

**Default value**

0x1e

**Description**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

**affinity-width****Type**

string

**Default value**

"4.8.8.8"

**Description**

A dotted quad indicating the bitwidth of fields at each affinity level.

**chip-id****Type**

int

**Default value**

0x0

**Description**

Chip ID when multichip operation is enabled.

**chip-select-affinity-level****Type**

int

**Default value**

0x3

**Description**

Affinity level 2 or 3 can be used for chip select.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**direct-lpi-support****Type**

bool

**Default value**

0x0

**Description**

Enable support for LPI operations through GICR registers.

**enable-multichip-operation****Type**

bool



**Default value**

0x0

**Description**

Enables multi-chip operation between Distributors in distributed GIC IRI.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**gicp-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

**gict-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

**max-cores-supported-by-GCI****Type**

int

**Default value**

0x8

**Description**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

**max-pe-on-chip****Type**

int

**Default value**

0x4

**Description**

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

**multichip-threaded-dgi****Type**

bool

**Default value**

0x1

**Description**

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if enable-multichip-operation is set.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**redistributor-group****Type**

string

**Default value**

-

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

### **redistributor-group-file**

#### **Type**

string

#### **Default value**

-

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

### **redistributor-power-managed-by-pwrr**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

GIC600 dedistributor power management is done by updating GICR\_PWRR register.

### **reg-base**

#### **Type**

int

#### **Default value**

0x2c010000

#### **Description**

GIC-600 base address.

### **reg-base-per-redistributor**

#### **Type**

string

#### **Default value**

{}

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

### 3.10.35 GIC600AE

GIC-600AE IRI implementation: Single chip validation/filter component variant limited to 265 PE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-979: IP revisions support**

Revision	Quality level
r0p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About GIC600AE

GIC600AE and GIC600AE\_Filter are minimal models of an Arm GIC-600AE Generic Interrupt Controller, suitable for single-chip or multichip systems, where multichip operation has been tested, with the following exceptions:

- LPI/LPI command is implemented but untested.
- Power operation is unimplemented.

They provide a simple configuration interface that allows designers to introduce GIC600AE-like functionality to their systems, while only implementing the architectural behavior, as defined by the GICv3 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC600AE and GIC600AE\_Filter have a `pvbuss_s` port for register accesses and a `pvbuss_m` port for the LPI-related traffic from redistributors and the ITS. The `pvbuss_m` port is also used to compose multichip operation.

In addition, the GIC600AE\_Filter variant has a `pvbuss_filtermiss_m` port, to which any transactions on the `pvbuss_s` port and not directed to a 4 KB page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC600AE variant.

It is recommended to use the GIC600AE variant in most cases.

Also, the RAS register is implemented and various RAS errors are reported through status registers. Not all the RAS errors are available because Fast Models does not model errors that can cause ECC errors. Fast Models supports error record 0 and error record 13+. FMU registers are also implemented and are accessible by software.



**Note**

- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.

- For writes to GITS\_TRANSLATE64R, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about GITS\_TRANSLATE64R, see [MSIRewriter](#).

## MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-980: MasterID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



**Note**

The `pvbus_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

## AE-specific features implemented

GIC600AE is a Functional Safety (FuSa) variant of GIC600. It has the following differences from GIC600:

- Both GIC600AE and GIC600 support RAS, but only GIC600AE supports the Fault Management Unit (FMU). This support is controlled by the `has-fmu` parameter which is true by default. The FMU resides in the Distributor and processes faults that are detected by the Safety Mechanisms from all blocks.
- In GIC600AE, the GIC reset pin is connected to FMU reset. Only reset changes the error record registers and `FMU_ERRGSR` to their reset values.
- In GIC600AE, the Safety Mechanism detects faults and forwards them to the FMU. The FMU forwards all errors to the Safety Island.
- The APB port has been added to GIC600AE for FuSa purposes. It does not exist on the GIC600.
- GIC600AE supports both Error Recovery Interrupt (ERI) and Fault Handling Interrupt (FHI).
- GIC600AE has the limitation that it only supports error injection through the `FMU_SMINJERR` register.

## Iris and MTI instances for GIC600AE

This model has the following Iris instances:

**Table 3-981: GIC600AE Iris instances**

InstanceName	ComponentName
GIC600AE	GIC_IRI

InstanceName	ComponentName
GIC600AE.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600AE.ITS0	GICv3InterruptTranslationService
GIC600AE.rd_0	GICv3RedistributorInternal
GIC600AE.rd_0_0	GICv3RedistributorInternal
GIC600AE.rd_0_0_0	GICv3RedistributorInternal
GIC600AE.rd_0_0_0_0	GICv3Redistributor
GIC600AE.rd_tl	GICv3Distributor

This model has the following MTI trace components:

**Table 3-982: GIC600AE MTI instances**

InstanceName	ComponentName
GIC600AE.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600AE.ITS0	GICv3InterruptTranslationService
GIC600AE.rd_0	GICv3RedistributorInternal
GIC600AE.rd_0_0	GICv3RedistributorInternal
GIC600AE.rd_0_0_0	GICv3RedistributorInternal
GIC600AE.rd_0_0_0_0	GICv3Redistributor
GIC600AE.rd_tl	GICv3Distributor

## Ports for GIC600AE

**Table 3-983: Ports**

Name	Protocol	Type	Description
apb_bus	PVBus	Slave	APB4 port to FMU, which will be connected from safety island for FuSa
chip_id	Value	Slave	chip_id port used for multichip operation
cpu_active_s[256]	Signal	Slave	CPUActive pins.
fm_u_error_int	Signal	Master	FuSa FMU error interrupt signal
fm_u_fault_int	Signal	Master	FuSa FMU fault interrupt signal
po_reset	Signal	Slave	Resets. This is used as a dbg_reset in TRM, and also be used for cold reset for PMU and FMU.
ppi_in_0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_10[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_100[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 106.

Name	Protocol	Type	Description
ppi_in_107[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 107.
ppi_in_108[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 109.
ppi_in_11[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_110[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 113.
ppi_in_114[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 114.
ppi_in_115[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 118.
ppi_in_119[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_12[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_120[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 120.
ppi_in_121[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 122.
ppi_in_123[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 123.
ppi_in_124[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 126.
ppi_in_127[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 128.
ppi_in_129[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_13[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_130[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_14[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_140[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 141.

Name	Protocol	Type	Description
ppi_in_142[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 142.
ppi_in_143[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 145.
ppi_in_146[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 149.
ppi_in_15[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_150[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 150.
ppi_in_151[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 153.
ppi_in_154[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 157.
ppi_in_158[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 159.
ppi_in_16[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_160[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 162.
ppi_in_163[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 164.
ppi_in_165[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_17[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_170[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 177.



Name	Protocol	Type	Description
ppi_in_178[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 178.
ppi_in_179[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_18[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_180[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 181.
ppi_in_182[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 184.
ppi_in_185[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 187.
ppi_in_188[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 189.
ppi_in_19[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_190[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 193.
ppi_in_194[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 194.
ppi_in_195[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 195.
ppi_in_196[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 199.
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_20[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_200[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_21[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_210[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 211.

Name	Protocol	Type	Description
ppi_in_212[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 212.
ppi_in_213[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 217.
ppi_in_218[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 219.
ppi_in_22[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_220[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 222.
ppi_in_223[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 223.
ppi_in_224[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 229.
ppi_in_23[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_230[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 230.
ppi_in_231[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 234.
ppi_in_235[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 236.
ppi_in_237[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_24[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_240[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 247.

Name	Protocol	Type	Description
ppi_in_248[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 248.
ppi_in_249[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_25[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_250[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 253.
ppi_in_254[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 254.
ppi_in_255[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_26[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_30[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_40[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_50[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 52.

Name	Protocol	Type	Description
ppi_in_53[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_60[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_70[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_8[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_80[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 88.

Name	Protocol	Type	Description
ppi_in_89[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_9[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 9.
ppi_in_90[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 95.
ppi_in_96[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 99.
pvbuss_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbuss_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Resets.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.

## Parameters for GIC600AE

### CPU-affinities

#### Type

string

#### Default value

0.0.0.0

#### Description

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

### CPU-affinities-file

#### Type

string

#### Default value

""

#### Description

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

**DS-behaviour****Type**

int

**Default value**

2

GICD\_CTLR.DS field behaviour

0 :RAZ/WI

1 :RAO/WI

2 :RW

**IIDR****Type**

int

**Default value**

0x300543b

**Description**

GICD\_IIDR, GICR\_IIDR and GITS\_IIDR value. Defaults to the latest revision.

**ITS-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of interrupt bits supported by ITS.

**ITS-collection-ID-bits****Type**

int

**Default value**

0x8

**Description**

Number of collection bits supported by ITS (optional parameter, 0 =&gt; 16bits support and GITS\_TYPER.CIL=0).

**ITS-count****Type**

int

**Default value**

0x1

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**ITS-device-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits supported for ITS device IDs.

**PPI-count****Type**

int

**Default value**

16

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31

**RAS-CFI-support****Type**

bool

**Default value**

0x0

**Description**

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

**RAS-FI-support****Type**

bool

**Default value**

0x0

**Description**

If true, fault handling interrupt is supported. Not supported otherwise.

**RAS-UE-support****Type**

bool

**Default value**

0x0

**Description**

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

**RAS-UI-support****Type**

bool

**Default value**

0x0

**Description**

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

**SPI-blocks****Type**

int

**Default value**

0x1e

**Description**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

**affinity-width****Type**

string

**Default value**

"4.8.8.8"



**Description**

A dotted quad indicating the bitwidth of fields at each affinity level.

**chip-id****Type**

int

**Default value**

0x0

**Description**

Chip ID when multichip operation is enabled.

**chip-select-affinity-level****Type**

int

**Default value**

0x3

**Description**

Affinity level 2 or 3 can be used for chip select.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**direct-lpi-support****Type**

bool

**Default value**

0x0

**Description**

Enable support for LPI operations through GICR registers.

**enable-multichip-operation****Type**

bool

**Default value**

0x0

**Description**

Enables multi-chip operation between Distributors in distributed GIC IRI.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**gicp-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

**gict-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

**max-cores-supported-by-GCI****Type**

int

**Default value**

0x8

**Description**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

**max-pe-on-chip****Type**

int

**Default value**

0x4

**Description**

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

**multichip-threaded-dgi****Type**

bool

**Default value**

0x1

**Description**

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if enable-multichip-operation is set.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**redistributor-group****Type**

string

**Default value**

-

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

### **redistributor-group-file**

#### **Type**

string

#### **Default value**

-

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

### **redistributor-power-managed-by-pwrr**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

GIC600 dedistributor power management is done by updating GICR\_PWRR register.

### **reg-base**

#### **Type**

int

#### **Default value**

0x2c010000

#### **Description**

GIC-600AE base address.

### **reg-base-per-redistributor**

#### **Type**

string

#### **Default value**

{}

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

### 3.10.36 GIC600AE\_Filter

GIC-600AE IRI implementation: Single chip validation/filter component variant limited to 265 PE.  
This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-984: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About GIC600AE\_Filter

GIC600AE and GIC600AE\_Filter are minimal models of an Arm GIC-600AE Generic Interrupt Controller, suitable for single-chip or multichip systems, where multichip operation has been tested, with the following exceptions:

- LPI/LPI command is implemented but untested.
- Power operation is unimplemented.

They provide a simple configuration interface that allows designers to introduce GIC600AE-like functionality to their systems, while only implementing the architectural behavior, as defined by the GICv3 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC600AE and GIC600AE\_Filter have a `pvbuss_s` port for register accesses and a `pvbuss_m` port for the LPI-related traffic from redistributors and the ITS. The `pvbuss_m` port is also used to compose multichip operation.

In addition, the GIC600AE\_Filter variant has a `pvbuss_filtermiss_m` port, to which any transactions on the `pvbuss_s` port and not directed to a 4 KB page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC600AE variant.

It is recommended to use the GIC600AE variant in most cases.

Also, the RAS register is implemented and various RAS errors are reported through status registers. Not all the RAS errors are available because Fast Models does not model errors that can cause ECC errors. Fast Models supports error record 0 and error record 13+. FMU registers are also implemented and are accessible by software.



**Note**

- Set the FASTSIM\_GIC\_MEMORY\_MAP environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.

- For writes to `GITS_TRANSLATE64R`, the `MSIRewriter` component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see [MSIRewriter](#).

## MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-985: MasterID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



Note

The `pvbus_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

## AE-specific features implemented

`GIC600AE_Filter` is a Functional Safety (FuSa) variant of `GIC600_Filter`. It has the following differences from `GIC600_Filter`:

- Both `GIC600AE_Filter` and `GIC600_Filter` support RAS, but only `GIC600AE_Filter` supports the Fault Management Unit (FMU). This support is controlled by the `has-fmu` parameter which is true by default. The FMU resides in the Distributor and processes faults that are detected by the Safety Mechanisms from all blocks.
- In `GIC600AE_Filter`, the GIC reset pin is connected to FMU reset. Only reset changes the error record registers and `FMU_ERRGSR` to their reset values.
- In `GIC600AE_Filter`, the Safety Mechanism detects faults and forwards them to the FMU. The FMU forwards all errors to the Safety Island.
- The APB port has been added to `GIC600AE_Filter` for FuSa purposes. It does not exist on the `GIC600_Filter`.
- `GIC600AE_Filter` supports both Error Recovery Interrupt (ERI) and Fault Handling Interrupt (FHI).
- `GIC600AE_Filter` has the limitation that it only supports error injection through the `FMU_SMINJERR` register.

## Iris and MTI instances for GIC600AE\_Filter

This model has the following Iris instances:

**Table 3-986: GIC600AE\_Filter Iris instances**

InstanceName	ComponentName
<code>GIC600AE_Filter</code>	<code>GIC_IRI</code>

InstanceName	ComponentName
GIC600AE_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600AE_Filter.ITS0	GICv3InterruptTranslationService
GIC600AE_Filter.rd_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0_0_0	GICv3Redistributor
GIC600AE_Filter.rd_tl	GICv3Distributor

This model has the following MTI trace components:

**Table 3-987: GIC600AE\_Filter MTI instances**

InstanceName	ComponentName
GIC600AE_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600AE_Filter.ITS0	GICv3InterruptTranslationService
GIC600AE_Filter.rd_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0_0_0	GICv3Redistributor
GIC600AE_Filter.rd_tl	GICv3Distributor

## Ports for GIC600AE\_Filter

**Table 3-988: Ports**

Name	Protocol	Type	Description
apb_bus	PVBus	Slave	APB4 port to FMU, which will be connected from safety island for FuSa
chip_id	Value	Slave	chip_id port used for multichip operation
cpu_active_s[256]	Signal	Slave	CPUActive pins.
fm_u_error_int	Signal	Master	FuSa FMU error interrupt signal
fm_u_fault_int	Signal	Master	FuSa FMU fault interrupt signal
po_reset	Signal	Slave	Reset.
ppi_in_0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_10[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_100[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 106.
ppi_in_107[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 107.

Name	Protocol	Type	Description
ppi_in_108[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 109.
ppi_in_11[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_110[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 113.
ppi_in_114[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 114.
ppi_in_115[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 118.
ppi_in_119[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_12[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_120[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 120.
ppi_in_121[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 122.
ppi_in_123[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 123.
ppi_in_124[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 126.
ppi_in_127[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 128.
ppi_in_129[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_13[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_130[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_14[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_140[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 141.
ppi_in_142[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 142.



Name	Protocol	Type	Description
ppi_in_143[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 145.
ppi_in_146[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 149.
ppi_in_15[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_150[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 150.
ppi_in_151[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 153.
ppi_in_154[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 157.
ppi_in_158[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 159.
ppi_in_16[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_160[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 162.
ppi_in_163[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 164.
ppi_in_165[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_17[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_170[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 177.
ppi_in_178[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 178.

Name	Protocol	Type	Description
ppi_in_179[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_18[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_180[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 181.
ppi_in_182[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 184.
ppi_in_185[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 187.
ppi_in_188[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 189.
ppi_in_19[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_190[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 193.
ppi_in_194[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 194.
ppi_in_195[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 195.
ppi_in_196[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 199.
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_20[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_200[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_21[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_210[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 211.
ppi_in_212[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 212.

Name	Protocol	Type	Description
ppi_in_213[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 217.
ppi_in_218[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 219.
ppi_in_22[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_220[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 222.
ppi_in_223[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 223.
ppi_in_224[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 229.
ppi_in_23[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_230[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 230.
ppi_in_231[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 234.
ppi_in_235[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 236.
ppi_in_237[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_24[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_240[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 247.
ppi_in_248[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 248.

Name	Protocol	Type	Description
ppi_in_249[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_25[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_250[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 253.
ppi_in_254[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 254.
ppi_in_255[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_26[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_30[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_40[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_50[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 53.

Name	Protocol	Type	Description
ppi_in_54[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_60[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_70[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_8[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_80[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 89.

Name	Protocol	Type	Description
ppi_in_9[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 9.
ppi_in_90[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 95.
ppi_in_96[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 99.
pvbus_filtermiss_m	PVBus	Master	Memory bus out. Transactions not filtered by the component.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Reset.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.

## Parameters for GIC600AE\_Filter

### CPU-affinities

#### Type

string

#### Default value

0.0.0.0

#### Description

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

### CPU-affinities-file

#### Type

string

#### Default value

""

#### Description

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

**DS-behaviour****Type**

int

**Default value**

2

GICD\_CTLR.DS field behaviour

0 :RAZ/WI

1 :RAO/WI

2 :RW

**IIDR****Type**

int

**Default value**

0x300543b

**Description**

GICD\_IIDR, GICR\_IIDR and GITS\_IIDR value. Defaults to the latest revision.

**ITS-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of interrupt bits supported by ITS.

**ITS-collection-ID-bits****Type**

int

**Default value**

0x8

**Description**

Number of collection bits supported by ITS (optional parameter, 0 =&gt; 16bits support and GITS\_TYPER.CIL=0).

**ITS-count****Type**

int

**Default value**

0x1

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**ITS-device-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits supported for ITS device IDs.

**PPI-count****Type**

int

**Default value**

16

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31

**RAS-CFI-support****Type**

bool

**Default value**

0x0

**Description**

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.



**RAS-FI-support****Type**

bool

**Default value**

0x0

**Description**

If true, fault handling interrupt is supported. Not supported otherwise.

**RAS-UE-support****Type**

bool

**Default value**

0x0

**Description**

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

**RAS-UI-support****Type**

bool

**Default value**

0x0

**Description**

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

**SPI-blocks****Type**

int

**Default value**

0x1e

**Description**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

**affinity-width****Type**

string

**Default value**

"4.8.8.8"

**Description**

A dotted quad indicating the bitwidth of fields at each affinity level.

**chip-id****Type**

int

**Default value**

0x0

**Description**

Chip ID when multichip operation is enabled.

**chip-select-affinity-level****Type**

int

**Default value**

0x3

**Description**

Affinity level 2 or 3 can be used for chip select.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**direct-lpi-support****Type**

bool

**Default value**

0x0

**Description**

Enable support for LPI operations through GICR registers.

**enable-multichip-operation****Type**

bool

**Default value**

0x0

**Description**

Enables multi-chip operation between Distributors in distributed GIC IRI.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**gicp-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

**gict-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

**max-cores-supported-by-GCI****Type**

int

**Default value**

0x8

**Description**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

**max-pe-on-chip****Type**

int

**Default value**

0x4

**Description**

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

**multichip-threaded-dgi****Type**

bool

**Default value**

0x1

**Description**

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if enable-multichip-operation is set.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**redistributor-group****Type**

string

**Default value**

-

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

### **redistributor-group-file**

#### **Type**

string

#### **Default value**

-

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

### **redistributor-power-managed-by-pwrr**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

GIC600 dedistributor power management is done by updating GICR\_PWRR register.

### **reg-base**

#### **Type**

int

#### **Default value**

0x2c010000

#### **Description**

GIC-600AE base address.

### **reg-base-per-redistributor**

#### **Type**

string

#### **Default value**

{}

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

### 3.10.37 GIC600\_Filter

GIC-600 IRI implementation: Single chip validation/filter component variant limited to 265 PE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-989: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About GIC600\_Filter

GIC600 and GIC600\_Filter are minimal models of an Arm GIC-600 Generic Interrupt Controller, suitable for single-chip or multichip systems, where multichip operation has been tested, with the following exceptions:

- LPI/LPI command is implemented but untested.
- Power operation is unimplemented.

They provide a simple configuration interface that allows designers to introduce GIC600-like functionality to their systems, while only implementing the architectural behavior, as defined by the GICv3 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC600 and GIC600\_Filter have a `pvbus_s` port for register accesses and a `pvbus_m` port for the LPI-related traffic from redistributors and the ITS. The `pvbus_m` port is also used to compose multichip operation.

In addition, the GIC600\_Filter variant has a `pvbus_filtermiss_m` port, to which any transactions on the `pvbus_s` port and not directed to a 4K page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC600 variant.

It is recommended to use the GIC600 variant in most cases.

Also, the RAS register is implemented and various RAS errors are reported through status registers. Not all the RAS errors are available because Fast Models does not model errors that can cause ECC errors. Fast Models supports error record 0 and error record 13+.



- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.

- For writes to GITS\_TRANSLATE64R, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about GITS\_TRANSLATE64R, see [MSIRewriter](#).

## MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-990: MasterID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



**Note**

The `pvbus_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

## Iris and MTI instances for GIC600\_Filter

This model has the following Iris instances:

**Table 3-991: GIC600\_Filter Iris instances**

InstanceName	ComponentName
GIC600_Filter	GIC_IRI
GIC600_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600_Filter.ITS0	GICv3InterruptTranslationService
GIC600_Filter.rd_0	GICv3RedistributorInternal
GIC600_Filter.rd_0_0	GICv3RedistributorInternal
GIC600_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC600_Filter.rd_0_0_0_0	GICv3Redistributor
GIC600_Filter.rd_tl	GICv3Distributor

This model has the following MTI trace components:

**Table 3-992: GIC600\_Filter MTI instances**

InstanceName	ComponentName
GIC600_Filter.GICV3_ProtocolChecker	<a href="#">GICv3ProtocolChecker</a>
GIC600_Filter.ITS0	<a href="#">GICv3InterruptTranslationService</a>
GIC600_Filter.rd_0	<a href="#">GICv3RedistributorInternal</a>
GIC600_Filter.rd_0_0	<a href="#">GICv3RedistributorInternal</a>
GIC600_Filter.rd_0_0_0	<a href="#">GICv3RedistributorInternal</a>

InstanceName	ComponentName
GIC600_Filter.rd_0_0_0_0	GICv3Redistributor
GIC600_Filter.rd_tl	GICv3Distributor

## Ports for GIC600\_Filter

**Table 3-993: Ports**

Name	Protocol	Type	Description
chip_id	Value	Slave	chip_id port which is valid from GIC600 r1p2. Writing to this port for prior GIC600 version will be ignored.
cpu_active_s[256]	Signal	Slave	CPUActive pins.
po_reset	Signal	Slave	Reset.
ppi_in_0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_10[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_100[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 106.
ppi_in_107[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 107.
ppi_in_108[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 109.
ppi_in_11[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_110[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 113.
ppi_in_114[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 114.
ppi_in_115[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 118.
ppi_in_119[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_12[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_120[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 120.
ppi_in_121[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 122.
ppi_in_123[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 123.



Name	Protocol	Type	Description
ppi_in_124[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 126.
ppi_in_127[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 128.
ppi_in_129[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_13[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_130[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_14[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_140[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 141.
ppi_in_142[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 142.
ppi_in_143[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 145.
ppi_in_146[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 149.
ppi_in_15[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_150[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 150.
ppi_in_151[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 153.
ppi_in_154[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 157.
ppi_in_158[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 159.

Name	Protocol	Type	Description
ppi_in_16[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_160[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 162.
ppi_in_163[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 164.
ppi_in_165[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_17[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_170[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 177.
ppi_in_178[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 178.
ppi_in_179[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_18[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_180[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 181.
ppi_in_182[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 184.
ppi_in_185[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 187.
ppi_in_188[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 189.
ppi_in_19[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_190[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 193.
ppi_in_194[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 194.

Name	Protocol	Type	Description
ppi_in_195[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 195.
ppi_in_196[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 199.
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_20[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_200[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_21[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_210[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 211.
ppi_in_212[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 212.
ppi_in_213[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 217.
ppi_in_218[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 219.
ppi_in_22[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_220[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 222.
ppi_in_223[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 223.
ppi_in_224[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 229.

Name	Protocol	Type	Description
ppi_in_23[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_230[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 230.
ppi_in_231[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 234.
ppi_in_235[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 236.
ppi_in_237[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_24[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_240[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 247.
ppi_in_248[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 248.
ppi_in_249[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_25[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_250[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 253.
ppi_in_254[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 254.
ppi_in_255[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_26[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_30[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 34.

Name	Protocol	Type	Description
ppi_in_35[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_40[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_50[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_60[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.

Name	Protocol	Type	Description
ppi_in_70[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_8[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_80[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_9[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 9.
ppi_in_90[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 95.
ppi_in_96[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 99.
pvbus_filtermiss_m	PVBus	Master	Memory bus out. Transactions not filtered by the component.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Reset.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.

## Parameters for GIC600\_Filter

### CPU-affinities

**Type**

string

**Default value**

0.0.0.0

**Description**

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

### CPU-affinities-file

**Type**

string

**Default value**

""

**Description**

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

### DS-behaviour

**Type**

int

**Default value**

2

GICD\_CTLR.DS field behaviour

0 :RAZ/WI

1 :RAO/WI

2 :RW

### IIDR

**Type**

int

**Default value**

0x201743b

**Description**

GICD\_IIDR and GICR\_IIDR value. Defaults to the latest revision.

**ITS-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of interrupt bits supported by ITS.

**ITS-collection-ID-bits****Type**

int

**Default value**

0x8

**Description**

Number of collection bits supported by ITS (optional parameter, 0 =&gt; 16bits support and GITS\_TYPER.CIL=0).

**ITS-count****Type**

int

**Default value**

0x1

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**ITS-device-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits supported for ITS device IDs.

**PPI-count****Type**

int

**Default value**

16

Selects the number of PPI available for each PE



8 :id22-27,29,30

**12**

id 20-31

**16**

id 16-31

### **RAS-CFI-support**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

### **RAS-FI-support**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

If true, fault handling interrupt is supported. Not supported otherwise.

### **RAS-UE-support**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

### **RAS-UI-support**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

**SPI-blocks****Type**

int

**Default value**

0x1e

**Description**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

**affinity-width****Type**

string

**Default value**

"4.8.8.8"

**Description**

A dotted quad indicating the bitwidth of fields at each affinity level.

**chip-id****Type**

int

**Default value**

0x0

**Description**

Chip ID when multichip operation is enabled.

**chip-select-affinity-level****Type**

int

**Default value**

0x3

**Description**

Affinity level 2 or 3 can be used for chip select.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**direct-lpi-support****Type**

bool

**Default value**

0x0

**Description**

Enable support for LPI operations through GICR registers.

**enable-multichip-operation****Type**

bool

**Default value**

0x0

**Description**

Enables multi-chip operation between Distributors in distributed GIC IRI.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**gicp-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

**gict-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

**max-cores-supported-by-GCI****Type**

int

**Default value**

0x8

**Description**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

**max-pe-on-chip****Type**

int

**Default value**

0x4

**Description**

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

**multichip-threaded-dgi****Type**

bool

**Default value**

0x1

**Description**

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if enable-multichip-operation is set.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**redistributor-group****Type**

string

**Default value**

-

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

**redistributor-group-file****Type**

string

**Default value**

-

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

**redistributor-power-managed-by-pwrr****Type**

bool

**Default value**

0x1

**Description**

GIC600 dedistributor power management is done by updating GICR\_PWRR register.

**reg-base****Type**

int

**Default value**

0x2c010000

Description

GIC-600 base address.

reg-base-per-redistributor

Type

string

Default value

{}

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the reg-base parameter (except that reg-base will still be used for the top-level redistributor). If reg-base-per-redistributor-file is specified, this parameter is ignored.

3.10.38 GIC625

GIC-625 IRI implementation: Single chip validation/filter component variant limited to 8 PE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-994: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About GIC625

GIC625 and GIC625\_Filter are Generic Interrupt Controllers that handle interrupts from peripherals to cores and interrupts between cores in a single cluster of up to 8 cores. They support the GICv3 and GICv3.1 architectures.

GIC625 supports the following features:

- Grouping cores in GCI.
- LPI is disabled.
- Power Management updates such as:
  - GCI grouping.
  - GCI RDPowerDown support
  - Updating the GCI ID for the cluster.

- GIC GICR\_PWRR register updates.
- Inversion of SPI and PPI inputs.
- Extended PPI.

**Note**

- Extended SPI is not supported.
- All implementation-specific registers and functionality are implemented.

## Iris and MTI instances for GIC625

This model has the following Iris instances:

**Table 3-995: GIC625 Iris instances**

InstanceName	ComponentName
GIC625	GIC_IRI
GIC625.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC625.rd_0	GICv3RedistributorInternal
GIC625.rd_0_0	GICv3RedistributorInternal
GIC625.rd_0_0_0	GICv3RedistributorInternal
GIC625.rd_0_0_0_0	GICv3Redistributor
GIC625.rd_t1	GICv3Distributor

This model has the following MTI trace components:

**Table 3-996: GIC625 MTI instances**

InstanceName	ComponentName
GIC625.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC625.rd_0	GICv3RedistributorInternal
GIC625.rd_0_0	GICv3RedistributorInternal
GIC625.rd_0_0_0	GICv3RedistributorInternal
GIC625.rd_0_0_0_0	GICv3Redistributor
GIC625.rd_t1	GICv3Distributor

## Ports for GIC625

**Table 3-997: Ports**

Name	Protocol	Type	Description
chip_id	Value	Slave	chip_id port used for multichip operation
cpu_active_s[8]	Signal	Slave	CPUActive pins.
po_reset	Signal	Slave	Resets. This is used as a dbg_reset in TRM, and also be used for cold reset for PMU and FMU.
ppi_in_0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.

Name	Protocol	Type	Description
ppi_in_1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[8]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Resets.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
wake_request[8]	Signal	Master	Power management outputs.

## Parameters for GIC625

### CPU-affinities

#### Type

string

#### Default value

0.0.0.0

#### Description

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

### CPU-affinities-file

#### Type

string

#### Default value

""

#### Description

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

### IIDR

#### Type

int

#### Default value

0x601043b



**Description**

GICD\_IIDR and GICR\_IIDR value.

**PPI-count****Type**

int

**Default value**

16

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31

**RAS-CFI-support****Type**

bool

**Default value**

0x0

**Description**

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

**RAS-FI-support****Type**

bool

**Default value**

0x0

**Description**

If true, fault handling interrupt is supported. Not supported otherwise.

**RAS-UE-support****Type**

bool

**Default value**

0x0

**Description**

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

**RAS-UI-support****Type**

bool

**Default value**

0x0

**Description**

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

**SPI-blocks****Type**

int

**Default value**

0x1e

**Description**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

**affinity-width****Type**

string

**Default value**

"4.8.8.8"

**Description**

A dotted quad indicating the bitwidth of fields at each affinity level.

**chip-select-affinity-level****Type**

int

**Default value**

0x3

**Description**

Affinity level 2 or 3 can be used for chip select.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**gicp-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

**gict-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

**has-two-security-states****Type**

bool

**Default value**

0x1

**Description**

If true, has two security states.

**max-cores-supported-by-GCI****Type**

int

**Default value**

0x8

**Description**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**redistributor-group****Type**

string

**Default value**

-

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

**redistributor-group-file****Type**

string

**Default value**

-

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

**redistributor-power-managed-by-pwrr**

**Type**

bool

**Default value**

0x1

**Description**

GIC600 dedistributor power management is done by updating GICR\_PWRR register.

**reg-base**

**Type**

int

**Default value**

0x2c010000

**Description**

GIC-625 base address.

**reg-base-per-redistributor**

**Type**

string

**Default value**

""

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

**3.10.39 GIC625\_Filter**

GIC-625 IRI implementation: Single chip validation/filter component variant limited to 8 PE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-998: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About GIC625\_Filter

GIC625 and GIC625\_Filter are Generic Interrupt Controllers that handle interrupts from peripherals to cores and interrupts between cores in a single cluster of up to 8 cores. They support the GICv3 and GICv3.1 architectures.

GIC625\_Filter supports the following features:

- Grouping cores in GCI.
- LPI is disabled.
- Power Management updates such as:
  - GCI grouping.
  - GCI RDPowerDown support
  - Updating the GCI ID for the cluster.
  - GCI GICR\_PWRR register updates.
- Inversion of SPI and PPI inputs.
- Extended PPI.



Note

- Extended SPI is not supported.
- All implementation-specific registers and functionality are implemented.

### Iris and MTI instances for GIC625\_Filter

This model has the following Iris instances:

**Table 3-999: GIC625\_Filter Iris instances**

InstanceName	ComponentName
GIC625_Filter	GIC_IRI
GIC625_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC625_Filter.rd_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0_0_0	GICv3Redistributor
GIC625_Filter.rd_t1	GICv3Distributor

This model has the following MTI trace components:

**Table 3-1000: GIC625\_Filter MTI instances**

InstanceName	ComponentName
GIC625_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker

InstanceName	ComponentName
GIC625_Filter.rd_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0_0_0	GICv3Redistributor
GIC625_Filter.rd_tl	GICv3Distributor

## Ports for GIC625\_Filter

**Table 3-1001: Ports**

Name	Protocol	Type	Description
chip_id	Value	Slave	chip_id port used for multichip operation
cpu_active_s[8]	Signal	Slave	CPUActive pins.
po_reset	Signal	Slave	Reset.
ppi_in_0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
pvbus_filtermiss_m	PVBus	Master	Memory bus out. Transactions not filtered by the component.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[8]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Reset.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
wake_request[8]	Signal	Master	Power management outputs.

## Parameters for GIC625\_Filter

### CPU-affinities

#### Type

string

#### Default value

0.0.0.0

#### Description

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

**CPU-affinities-file****Type**

string

**Default value**

""

**Description**

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

**IIDR****Type**

int

**Default value**

0x601043b

**Description**

GICD\_IIDR and GICR\_IIDR value.

**PPI-count****Type**

int

**Default value**

16

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31

**RAS-CFI-support****Type**

bool

**Default value**

0x0

**Description**

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.



**RAS-FI-support****Type**

bool

**Default value**

0x0

**Description**

If true, fault handling interrupt is supported. Not supported otherwise.

**RAS-UE-support****Type**

bool

**Default value**

0x0

**Description**

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

**RAS-UI-support****Type**

bool

**Default value**

0x0

**Description**

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

**SPI-blocks****Type**

int

**Default value**

0x1e

**Description**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

**affinity-width****Type**

string

**Default value**

"4.8.8.8"

**Description**

A dotted quad indicating the bitwidth of fields at each affinity level.

**chip-select-affinity-level****Type**

int

**Default value**

0x3

**Description**

Affinity level 2 or 3 can be used for chip select.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**gicp-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

**gict-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

**has-two-security-states****Type**

bool

**Default value**

0x1

**Description**

If true, has two security states.

**max-cores-supported-by-GCI****Type**

int

**Default value**

0x8

**Description**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**redistributor-group****Type**

string

**Default value**

-

Redistributor grouping information with affinity as JSON :

```
{
  "0":
    ["0.0.0.0",
     "0.0.0.1"],
```

```
"1":
  ["0.0.1.0",
   "0.0.1.1"]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

### **redistributor-group-file**

#### **Type**

string

#### **Default value**

-

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

### **redistributor-power-managed-by-pwrr**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

GIC600 dedistributor power management is done by updating GICR\_PWRR register.

### **reg-base**

#### **Type**

int

#### **Default value**

0x2c010000

#### **Description**

GIC-625 base address.

### **reg-base-per-redistributor**

#### **Type**

string

#### **Default value**

{}

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

### 3.10.40 GIC700

GIC-700 IRI implementation: Single/multi chip validation/filter component variant limited to 64 PE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1002: IP revisions support**

Revision	Quality level
r0p0	Full support
r1p0	Full support
r2p0	Full support
r3p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Changes in 11.29.19

Parameters added:

- `has_nmi`

#### About GIC700

GIC700 and GIC700\_Filter are minimal models of an Arm® GIC-700 Generic Interrupt Controller, suitable for single-chip or multichip systems. They support the following multichip operations:

- Multichip configuration through distributor registers
- SPI
- SGI
- Physical LPI
- Physical LPI command
- Virtual LPI
- Virtual LPI command

The models provide a simple configuration interface that allows designers to introduce GIC700-like functionality to systems, while only implementing the architectural behavior, as defined by the GICv3/GICv4 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC700 and GIC700\_Filter have a `pvbuss_s` port for register accesses and a `pvbuss_m` port for the LPI-related traffic from redistributors and the ITS.

In addition, the GIC700\_Filter variant has a `pvbuss_filtermiss_m` port, to which any transactions on the `pvbuss_s` port and not directed to a 4 KB page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC700 variant.

GIC700 and GIC700\_Filter support an extended interrupt range, with 64 more PPIs and 1024 more SPIs. To program additional interrupts, refer to the GICv3/GICv4 specification.

GIC700 and GIC700\_Filter support direct injection of virtual interrupts. This reduces the overhead of trapping to hypervisor for the generation of virtual interrupts. In the GICv3/GICv4 specification, direct injection is supported in two different versions, GICv4.0 and GICv4.1. GIC700 and GIC700\_Filter support GICv4.1.

It is recommended to use GIC700 instead of GIC700\_Filter in most cases.

The GIC700 model supports multiple revisions of the IP. By default, its behavior is set to the latest revision supported. Parameters can be used to enable or disable features that are specific to earlier revisions. The LCA (Local Cross-chip Addressing) feature which is added in r2p0 is disabled by default and is enabled by setting the parameter `enable-local-cross-chip-addressing` to true. The model's registers follow the specification for the latest IP revision supported.



Note

- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- For writes to `GITS_TRANSLATE64R`, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see [MSIRewriter](#).

## Limitations

- The following multichip operations are not yet supported:
  - vPE control.
  - Power control.
- There is no support for the VMOV command.
- Limited testing has been performed for multichip LPI operations, so the model might contain defects.
- One ITS is supported for each GIC for multichip operations.
- There is no support for VICM.

## MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-1003: MasterID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used

**Note**

The `pvbus_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

**Iris and MTI instances for GIC700**

This model has the following Iris instances:

**Table 3-1004: GIC700 Iris instances**

InstanceName	ComponentName
GIC700	GIC_IRI
GIC700.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC700.ITS0	GICv4InterruptTranslationService
GIC700.ITS0.bus_subordinate	PVBusSlave
GIC700.rd_0	GICv4RedistributorInternal
GIC700.rd_0_0	GICv4RedistributorInternal
GIC700.rd_0_0_0	GICv4RedistributorInternal
GIC700.rd_0_0_0_0	GICv3Redistributor
GIC700.rd_t1	GICv3Distributor

This model has the following MTI trace components:

**Table 3-1005: GIC700 MTI instances**

InstanceName	ComponentName
GIC700.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC700.ITS0	GICv4InterruptTranslationService
GIC700.ITS0.bus_subordinate	PVBusSlave
GIC700.rd_0	GICv4RedistributorInternal
GIC700.rd_0_0	GICv4RedistributorInternal
GIC700.rd_0_0_0	GICv4RedistributorInternal
GIC700.rd_0_0_0_0	GICv3Redistributor
GIC700.rd_t1	GICv3Distributor

## Ports for GIC700

Table 3-1006: Ports

Name	Protocol	Type	Description
axi_stream_msi_s[32]	PVBus	Slave	AXI Stream Interface ports for MSI (message-signalled interrupt) translation, in GIC-700. Typically the SMMU's TCU connects to this port for MSI.
chip_id	Value	Slave	chip_id port for multichip operation
cpu_active_s[256]	Signal	Slave	CPUActive pins.
cpu_wake_request[256]	Signal	Master	-
extended_ppi_in_0[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
extended_ppi_in_1[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
extended_ppi_in_10[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
extended_ppi_in_100[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100
extended_ppi_in_101[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101
extended_ppi_in_102[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102
extended_ppi_in_103[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103
extended_ppi_in_104[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104
extended_ppi_in_105[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105
extended_ppi_in_106[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106
extended_ppi_in_107[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107
extended_ppi_in_108[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108
extended_ppi_in_109[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109
extended_ppi_in_11[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
extended_ppi_in_110[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110
extended_ppi_in_111[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111
extended_ppi_in_112[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112
extended_ppi_in_113[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113
extended_ppi_in_114[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114



Name	Protocol	Type	Description
extended_ppi_in_115[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115
extended_ppi_in_116[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116
extended_ppi_in_117[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117
extended_ppi_in_118[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118
extended_ppi_in_119[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119
extended_ppi_in_12[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_120[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120
extended_ppi_in_121[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121
extended_ppi_in_122[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122
extended_ppi_in_123[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123
extended_ppi_in_124[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124
extended_ppi_in_125[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125
extended_ppi_in_126[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126
extended_ppi_in_127[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127
extended_ppi_in_128[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128
extended_ppi_in_129[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129
extended_ppi_in_13[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.
extended_ppi_in_130[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130
extended_ppi_in_131[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131
extended_ppi_in_132[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132
extended_ppi_in_133[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133
extended_ppi_in_134[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134
extended_ppi_in_135[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135

Name	Protocol	Type	Description
extended_ppi_in_136[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136
extended_ppi_in_137[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137
extended_ppi_in_138[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138
extended_ppi_in_139[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139
extended_ppi_in_14[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.
extended_ppi_in_140[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140
extended_ppi_in_141[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141
extended_ppi_in_142[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142
extended_ppi_in_143[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143
extended_ppi_in_144[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144
extended_ppi_in_145[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145
extended_ppi_in_146[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146
extended_ppi_in_147[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147
extended_ppi_in_148[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148
extended_ppi_in_149[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149
extended_ppi_in_15[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_150[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150
extended_ppi_in_151[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151
extended_ppi_in_152[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152
extended_ppi_in_153[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153
extended_ppi_in_154[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154
extended_ppi_in_155[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155
extended_ppi_in_156[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156

Name	Protocol	Type	Description
extended_ppi_in_157[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157
extended_ppi_in_158[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158
extended_ppi_in_159[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159
extended_ppi_in_16[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_160[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160
extended_ppi_in_161[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161
extended_ppi_in_162[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162
extended_ppi_in_163[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163
extended_ppi_in_164[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164
extended_ppi_in_165[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165
extended_ppi_in_166[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166
extended_ppi_in_167[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167
extended_ppi_in_168[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168
extended_ppi_in_169[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169
extended_ppi_in_17[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_170[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170
extended_ppi_in_171[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171
extended_ppi_in_172[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172
extended_ppi_in_173[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173
extended_ppi_in_174[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174
extended_ppi_in_175[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175
extended_ppi_in_176[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176
extended_ppi_in_177[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177

Name	Protocol	Type	Description
extended_ppi_in_178[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178
extended_ppi_in_179[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179
extended_ppi_in_18[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_180[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180
extended_ppi_in_181[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181
extended_ppi_in_182[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182
extended_ppi_in_183[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183
extended_ppi_in_184[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184
extended_ppi_in_185[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185
extended_ppi_in_186[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186
extended_ppi_in_187[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187
extended_ppi_in_188[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188
extended_ppi_in_189[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189
extended_ppi_in_19[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.
extended_ppi_in_190[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190
extended_ppi_in_191[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191
extended_ppi_in_192[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192
extended_ppi_in_193[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193
extended_ppi_in_194[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194
extended_ppi_in_195[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195
extended_ppi_in_196[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196
extended_ppi_in_197[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197
extended_ppi_in_198[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198

Name	Protocol	Type	Description
extended_ppi_in_199[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199
extended_ppi_in_2[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_20[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_200[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200
extended_ppi_in_201[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201
extended_ppi_in_202[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202
extended_ppi_in_203[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203
extended_ppi_in_204[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204
extended_ppi_in_205[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205
extended_ppi_in_206[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206
extended_ppi_in_207[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207
extended_ppi_in_208[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208
extended_ppi_in_209[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209
extended_ppi_in_21[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.
extended_ppi_in_210[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210
extended_ppi_in_211[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211
extended_ppi_in_212[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212
extended_ppi_in_213[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213
extended_ppi_in_214[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214
extended_ppi_in_215[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215
extended_ppi_in_216[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216
extended_ppi_in_217[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217
extended_ppi_in_218[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218

Name	Protocol	Type	Description
extended_ppi_in_219[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219
extended_ppi_in_22[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.
extended_ppi_in_220[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220
extended_ppi_in_221[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221
extended_ppi_in_222[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222
extended_ppi_in_223[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223
extended_ppi_in_224[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224
extended_ppi_in_225[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225
extended_ppi_in_226[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226
extended_ppi_in_227[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227
extended_ppi_in_228[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228
extended_ppi_in_229[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229
extended_ppi_in_23[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_230[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230
extended_ppi_in_231[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231
extended_ppi_in_232[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232
extended_ppi_in_233[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233
extended_ppi_in_234[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234
extended_ppi_in_235[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235
extended_ppi_in_236[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236
extended_ppi_in_237[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237
extended_ppi_in_238[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238
extended_ppi_in_239[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239

Name	Protocol	Type	Description
extended_ppi_in_24[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.
extended_ppi_in_240[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240
extended_ppi_in_241[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241
extended_ppi_in_242[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242
extended_ppi_in_243[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243
extended_ppi_in_244[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244
extended_ppi_in_245[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245
extended_ppi_in_246[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246
extended_ppi_in_247[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247
extended_ppi_in_248[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248
extended_ppi_in_249[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249
extended_ppi_in_25[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_250[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250
extended_ppi_in_251[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251
extended_ppi_in_252[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252
extended_ppi_in_253[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253
extended_ppi_in_254[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254
extended_ppi_in_255[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255
extended_ppi_in_26[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.
extended_ppi_in_28[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_3[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.

Name	Protocol	Type	Description
extended_ppi_in_30[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.
extended_ppi_in_31[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.
extended_ppi_in_32[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.
extended_ppi_in_35[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.
extended_ppi_in_36[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_4[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_40[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.
extended_ppi_in_42[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.
extended_ppi_in_43[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.
extended_ppi_in_47[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.
extended_ppi_in_49[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_5[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_50[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.



Name	Protocol	Type	Description
extended_ppi_in_51[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.
extended_ppi_in_52[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.
extended_ppi_in_54[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.
extended_ppi_in_56[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_6[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_60[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.
extended_ppi_in_64[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64
extended_ppi_in_65[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65
extended_ppi_in_66[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66
extended_ppi_in_67[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67
extended_ppi_in_68[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68
extended_ppi_in_69[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69
extended_ppi_in_7[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_70[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70
extended_ppi_in_71[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71

Name	Protocol	Type	Description
extended_ppi_in_72[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72
extended_ppi_in_73[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73
extended_ppi_in_74[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74
extended_ppi_in_75[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75
extended_ppi_in_76[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76
extended_ppi_in_77[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77
extended_ppi_in_78[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78
extended_ppi_in_79[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79
extended_ppi_in_8[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.
extended_ppi_in_80[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80
extended_ppi_in_81[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81
extended_ppi_in_82[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82
extended_ppi_in_83[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83
extended_ppi_in_84[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84
extended_ppi_in_85[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85
extended_ppi_in_86[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86
extended_ppi_in_87[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87
extended_ppi_in_88[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88
extended_ppi_in_89[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89
extended_ppi_in_9[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.
extended_ppi_in_90[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90
extended_ppi_in_91[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91
extended_ppi_in_92[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92

Name	Protocol	Type	Description
extended_ppi_in_93[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93
extended_ppi_in_94[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94
extended_ppi_in_95[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95
extended_ppi_in_96[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96
extended_ppi_in_97[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97
extended_ppi_in_98[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98
extended_ppi_in_99[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99
extended_spi_in[1024]	Signal	Slave	Extended Shared peripheral interrupts.
icdrt_out	PVBus	Master	AXI Stream Interface. This is used only when local cross chip addressing is enabled at the moment
icrdt_in	PVBus	Slave	-
po_reset	Signal	Slave	Resets.
ppi_in_0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_10[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_100[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 100
ppi_in_101[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 101
ppi_in_102[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 102
ppi_in_103[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 103
ppi_in_104[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 104
ppi_in_105[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 105
ppi_in_106[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 106
ppi_in_107[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 107
ppi_in_108[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 108
ppi_in_109[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 109
ppi_in_11[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_110[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 110
ppi_in_111[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 111
ppi_in_112[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 112
ppi_in_113[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 113
ppi_in_114[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 114
ppi_in_115[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 115
ppi_in_116[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 116
ppi_in_117[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 117

Name	Protocol	Type	Description
ppi_in_118[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 118
ppi_in_119[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 119
ppi_in_12[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_120[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 120
ppi_in_121[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 121
ppi_in_122[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 122
ppi_in_123[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 123
ppi_in_124[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 124
ppi_in_125[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 125
ppi_in_126[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 126
ppi_in_127[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 127
ppi_in_128[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 128
ppi_in_129[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 129
ppi_in_13[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_130[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 130
ppi_in_131[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 131
ppi_in_132[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 132
ppi_in_133[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 133
ppi_in_134[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 134
ppi_in_135[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 135
ppi_in_136[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 136
ppi_in_137[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 137
ppi_in_138[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 138
ppi_in_139[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 139
ppi_in_14[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_140[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 140
ppi_in_141[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 141
ppi_in_142[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 142
ppi_in_143[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 143
ppi_in_144[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 144
ppi_in_145[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 145
ppi_in_146[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 146
ppi_in_147[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 147
ppi_in_148[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 148
ppi_in_149[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 149
ppi_in_15[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_150[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 150
ppi_in_151[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 151
ppi_in_152[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 152

Name	Protocol	Type	Description
ppi_in_153[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 153
ppi_in_154[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 154
ppi_in_155[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 155
ppi_in_156[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 156
ppi_in_157[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 157
ppi_in_158[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 158
ppi_in_159[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 159
ppi_in_16[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_160[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 160
ppi_in_161[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 161
ppi_in_162[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 162
ppi_in_163[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 163
ppi_in_164[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 164
ppi_in_165[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 165
ppi_in_166[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 166
ppi_in_167[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 167
ppi_in_168[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 168
ppi_in_169[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 169
ppi_in_17[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_170[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 170
ppi_in_171[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 171
ppi_in_172[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 172
ppi_in_173[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 173
ppi_in_174[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 174
ppi_in_175[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 175
ppi_in_176[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 176
ppi_in_177[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 177
ppi_in_178[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 178
ppi_in_179[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 179
ppi_in_18[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_180[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 180
ppi_in_181[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 181
ppi_in_182[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 182
ppi_in_183[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 183
ppi_in_184[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 184
ppi_in_185[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 185
ppi_in_186[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 186
ppi_in_187[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 187
ppi_in_188[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 188

Name	Protocol	Type	Description
ppi_in_189[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 189
ppi_in_19[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_190[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 190
ppi_in_191[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 191
ppi_in_192[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 192
ppi_in_193[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 193
ppi_in_194[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 194
ppi_in_195[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 195
ppi_in_196[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 196
ppi_in_197[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 197
ppi_in_198[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 198
ppi_in_199[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 199
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_20[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_200[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 200
ppi_in_201[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 201
ppi_in_202[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 202
ppi_in_203[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 203
ppi_in_204[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 204
ppi_in_205[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 205
ppi_in_206[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 206
ppi_in_207[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 207
ppi_in_208[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 208
ppi_in_209[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 209
ppi_in_21[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_210[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 210
ppi_in_211[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 211
ppi_in_212[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 212
ppi_in_213[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 213
ppi_in_214[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 214
ppi_in_215[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 215
ppi_in_216[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 216
ppi_in_217[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 217
ppi_in_218[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 218
ppi_in_219[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 219
ppi_in_22[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_220[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 220
ppi_in_221[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 221
ppi_in_222[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 222

Name	Protocol	Type	Description
ppi_in_223[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 223
ppi_in_224[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 224
ppi_in_225[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 225
ppi_in_226[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 226
ppi_in_227[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 227
ppi_in_228[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 228
ppi_in_229[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 229
ppi_in_23[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_230[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 230
ppi_in_231[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 231
ppi_in_232[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 232
ppi_in_233[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 233
ppi_in_234[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 234
ppi_in_235[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 235
ppi_in_236[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 236
ppi_in_237[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 237
ppi_in_238[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 238
ppi_in_239[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 239
ppi_in_24[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_240[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 240
ppi_in_241[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 241
ppi_in_242[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 242
ppi_in_243[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 243
ppi_in_244[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 244
ppi_in_245[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 245
ppi_in_246[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 246
ppi_in_247[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 247
ppi_in_248[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 248
ppi_in_249[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 249
ppi_in_25[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_250[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 250
ppi_in_251[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 251
ppi_in_252[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 252
ppi_in_253[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 253
ppi_in_254[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 254
ppi_in_255[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 255
ppi_in_26[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 28.



Name	Protocol	Type	Description
ppi_in_29[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_30[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_40[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_50[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_60[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 63.



Name	Protocol	Type	Description
ppi_in_64[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 64
ppi_in_65[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 65
ppi_in_66[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 66
ppi_in_67[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 67
ppi_in_68[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 68
ppi_in_69[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 69
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_70[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 70
ppi_in_71[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 71
ppi_in_72[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 72
ppi_in_73[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 73
ppi_in_74[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 74
ppi_in_75[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 75
ppi_in_76[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 76
ppi_in_77[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 77
ppi_in_78[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 78
ppi_in_79[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 79
ppi_in_8[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_80[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 80
ppi_in_81[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 81
ppi_in_82[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 82
ppi_in_83[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 83
ppi_in_84[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 84
ppi_in_85[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 85
ppi_in_86[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 86
ppi_in_87[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 87
ppi_in_88[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 88
ppi_in_89[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 89
ppi_in_9[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 9.
ppi_in_90[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 90
ppi_in_91[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 91
ppi_in_92[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 92
ppi_in_93[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 93
ppi_in_94[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 94
ppi_in_95[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 95
ppi_in_96[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 96
ppi_in_97[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 97
ppi_in_98[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 98
ppi_in_99[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 99

Name	Protocol	Type	Description
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Resets.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.

## Parameters for GIC700

### CPU-affinities

#### Type

string

#### Default value

0.0.0.0

#### Description

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

### CPU-affinities-file

#### Type

string

#### Default value

""

#### Description

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

### GICD\_CTLR\_DS-1-means-secure-only

#### Type

bool

#### Default value

0x0

#### Description

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

### GICD\_TYPER2

#### Type

int

**Default value**

0x0

**Description**

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

**IIDR****Type**

int

**Default value**

0x403043b

**Description**

GICD\_IIDR and GICR\_IIDR value.

**ITS-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of interrupt bits supported by ITS.

**ITS-collection-ID-bits****Type**

int

**Default value**

0x8

**Description**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

**ITS-count****Type**

int

**Default value**

0x1

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**ITS-device-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits supported for ITS device IDs.

**ITS-enable-itt-address-verification****Type**

bool

**Default value**

0x0

**Description**

If true, a transaction will be sent to ITT Address for verification.

**ITS-hardware-collection-count****Type**

int

**Default value**

0x0

**Description**

Number of hardware collections held exclusively in the ITS.

**ITS-shared-vPE-table****Type**

int

**Default value**

0x0

**Description**

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

**ITS-vmovp-bit****Type**

bool

**Default value**

0x0

**Description**

Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

**PPI-count****Type**

int

**Default value**

16

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31

**SPI-blocks****Type**

int

**Default value**

0x3e

**Description**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

**add-output-cpu-wake-request-signal-from-redistributor****Type**

bool

**Default value**

0x0

**Description**

if true, the redistributor will have the output signal cpu\_wake\_request from GIC to DSU and if false, the signals are not added to the redistributor.

**affinity-width****Type**

string

**Default value**

"4.8.8.8"

**Description**

A dotted quad indicating the bitwidth of fields at each affinity level.

**allow-LPIEN-clear****Type**

bool

**Default value**

0x1

**Description**

Allow RW behaviour on GICR\_CTLR.LPIEN instead of set once.

**chip-count****Type**

int

**Default value**

0x10

**Description**

The total number of chips supported.

**chip-id****Type**

int

**Default value**

0x0

**Description**

Chip ID when multichip operation is enabled.

**chip-select-affinity-level****Type**

int

**Default value**

0x3

**Description**

Affinity level 2 or 3 can be used for chip select.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**common-vPE-table-affinity****Type**

string

**Default value**

""

**Description**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s) (where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on). Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when has-gicv4.1 is true.

**consolidators****Type**

string

**Default value**

""

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form:

```
'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1'
```

For example:

```
'0x3f100000:64:4096, 0x3f200000:64:4224'
```

The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

**cross-chip-AMBA-is-ACE****Type**

bool

**Default value**

0x0

**Description**

Indicates the AMBA protocol that the cross-chip interface uses. If true, the cross-chip interface uses the ACE5-Lite protocol. Otherwise, it uses the AXI5-Stream protocol. This parameter is meaningful only if enable-multichip-operation is set.

**enable-local-cross-chip-addressing****Type**

bool

**Default value**

0x0

**Description**

If true, each distributor in a multichip system will use routing table address (ADDR) values programmed by local software, and will not be overwritten by the default chip. Otherwise, all the distributors will share the same routing table address values programmed in the default chip. This parameter is valid only if enable-multichip-operation is set.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**extended-ppi-count****Type**

int

**Default value**

0x40

**Description**

Number of extended PPI supported.

**gicp-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

**gict-allow-ns-reset****Type**

bool



**Default value**

0x1

**Description**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

**has-gicv4.1****Type**

bool

**Default value**

0x1

**Description**

Enable GICv4.1 functionality; when false the component is inactive.

**has-two-security-states****Type**

bool

**Default value**

0x1

**Description**

If true, has two security states.

**has\_nmi****Type**

bool

**Default value**

0x0

**Description**

Enable support for Non-maskable Interrupts (NMIs). (FEAT\_GICv3\_NMI).

**max-cores-supported-by-GCI****Type**

int

**Default value**

0x8

**Description**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

**max-pe-on-chip****Type**

int

**Default value**

0x4

**Description**

Maximum number of cores on any single chip. This will be used to identify the target chip and core.

**multichip-threaded-dgi****Type**

bool

**Default value**

0x1

**Description**

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if enable-multichip-operation is set.

**output\_attributes****Type**

string

**Default value**

"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID,  
ExtendedID[38]=MPAM\_NS"

**Description**

User-defined transform to be applied to bus attributes like MasterID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**prog-mpidr****Type**

unsigned

**Default value**

0

Whether software or hardware can remove cores from a GIC configuration.

**0**

none

**1**

prog - Secure software to remove cores during the boot up of a system.

**2**

strap - enables hardware to remove cores as GIC exits reset

**redistributor-group****Type**

string

**Default value**

-

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

**redistributor-group-file****Type**

string

**Default value**

-

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

**redistributor-power-managed-by-pwrr****Type**

bool

**Default value**  
0x1

**Description**  
GIC-700 dedistributor power management is done by updating GICR\_PWRR register.

**reg-base**

**Type**  
int

**Default value**  
0x2c010000

**Description**  
GIC-700 base address.

3.10.41 GIC700\_Filter

GIC-700 IRI implementation: Single/multi chip validation/filter component variant limited to 64 PE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1007: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Changes in 11.29.19**

Parameters added:

- has\_nmi

**About GIC700\_Filter**

GIC700 and GIC700\_Filter are minimal models of an Arm&reg GIC-700 Generic Interrupt Controller, suitable for single-chip or multichip systems. They support the following multichip operations:

- Multichip configuration through distributor registers.
- SPI.
- SGL.
- Physical LPI.
- Physical LPI command.
- Virtual LPI.
- Virtual LPI command.

The models provide a simple configuration interface that allows designers to introduce GIC700-like functionality to systems, while only implementing the architectural behavior, as defined by the GICv3/GICv4 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC700 and GIC700\_Filter have a `pvbus_s` port for register accesses and a `pvbus_m` port for the LPI-related traffic from redistributors and the ITS.

In addition, the GIC700\_Filter variant has a `pvbus_filtermiss_m` port, to which any transactions on the `pvbus_s` port and not directed to a 4 KB page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC700 variant.

GIC700 and GIC700\_Filter support an extended interrupt range, with 64 more PPIs and 1024 more SPIs. To program additional interrupts, refer to the GICv3/GICv4 specification.

GIC700 and GIC700\_Filter support direct injection of virtual interrupts. This reduces the overhead of trapping to hypervisor for the generation of virtual interrupts. In the GICv3/GICv4 specification, direct injection is supported in two different versions, GICv4.0 and GICv4.1. GIC700 and GIC700\_Filter support GICv4.1.

It is recommended to use GIC700 instead of GIC700\_Filter in most cases.

The GIC700 model supports multiple revisions of the IP. By default, its behavior is set to the latest revision supported. Parameters can be used to enable or disable features that are specific to earlier revisions. The LCA (Local Cross-chip Addressing) feature which is added in r2p0 is disabled by default and is enabled by setting the parameter `enable-local-cross-chip-addressing` to true. The model's registers follow the specification for the latest IP revision supported.



Note

- Set the FASTSIM\_GIC\_MEMORY\_MAP environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- For writes to `GITS_TRANSLATE64R`, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see [MSIRewriter](#).

## Limitations

- The following multichip operations are not yet supported:
  - vPE control.
  - Power control.
- There is no support for the VMOV command.
- Limited testing has been performed for multichip LPI operations, so the model might contain defects.
- One ITS is supported for each GIC for multichip operations.

## MasterID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `MasterID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-1008: MasterID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
MasterID	31:0	Master ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



Note

The `pvbus_m` port does not use `MasterID`, `ExtendedID`, or `UserFlags`.

## Iris and MTI instances for GIC700\_Filter

This model has the following Iris instances:

**Table 3-1009: GIC700\_Filter Iris instances**

InstanceName	ComponentName
GIC700_Filter	GIC_IRI
GIC700_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC700_Filter.ITS0	GICv4InterruptTranslationService
GIC700_Filter.ITS0.bus_subordinate	PVBusSlave
GIC700_Filter.rd_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0_0_0	GICv3Redistributor
GIC700_Filter.rd_t1	GICv3Distributor

This model has the following MTI trace components:

**Table 3-1010: GIC700\_Filter MTI instances**

InstanceName	ComponentName
GIC700_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC700_Filter.ITS0	GICv4InterruptTranslationService
GIC700_Filter.ITS0.bus_subordinate	PVBusSlave
GIC700_Filter.rd_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0_0_0	GICv3Redistributor
GIC700_Filter.rd_t1	GICv3Distributor

## Ports for GIC700\_Filter

Table 3-1011: Ports

Name	Protocol	Type	Description
axi_stream_msi_s[32]	PVBus	Slave	AXI Stream Interface ports for MSI (message-signalled interrupt) translation, in GIC-700. Typically the SMMU's TCU connects to this port for MSI.
chip_id	Value	Slave	chip_id port for multichip operation
cpu_active_s[256]	Signal	Slave	CPUActive pins.
extended_ppi_in_0[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
extended_ppi_in_1[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
extended_ppi_in_10[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
extended_ppi_in_100[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100
extended_ppi_in_101[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101
extended_ppi_in_102[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102
extended_ppi_in_103[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103
extended_ppi_in_104[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104
extended_ppi_in_105[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105
extended_ppi_in_106[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106
extended_ppi_in_107[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107
extended_ppi_in_108[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108
extended_ppi_in_109[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109
extended_ppi_in_11[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
extended_ppi_in_110[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110
extended_ppi_in_111[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111
extended_ppi_in_112[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112
extended_ppi_in_113[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113
extended_ppi_in_114[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114
extended_ppi_in_115[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115

Name	Protocol	Type	Description
extended_ppi_in_116[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116
extended_ppi_in_117[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117
extended_ppi_in_118[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118
extended_ppi_in_119[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119
extended_ppi_in_12[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_120[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120
extended_ppi_in_121[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121
extended_ppi_in_122[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122
extended_ppi_in_123[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123
extended_ppi_in_124[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124
extended_ppi_in_125[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125
extended_ppi_in_126[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126
extended_ppi_in_127[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127
extended_ppi_in_128[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128
extended_ppi_in_129[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129
extended_ppi_in_13[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.
extended_ppi_in_130[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130
extended_ppi_in_131[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131
extended_ppi_in_132[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132
extended_ppi_in_133[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133
extended_ppi_in_134[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134
extended_ppi_in_135[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135
extended_ppi_in_136[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136



Name	Protocol	Type	Description
extended_ppi_in_137[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137
extended_ppi_in_138[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138
extended_ppi_in_139[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139
extended_ppi_in_14[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.
extended_ppi_in_140[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140
extended_ppi_in_141[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141
extended_ppi_in_142[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142
extended_ppi_in_143[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143
extended_ppi_in_144[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144
extended_ppi_in_145[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145
extended_ppi_in_146[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146
extended_ppi_in_147[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147
extended_ppi_in_148[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148
extended_ppi_in_149[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149
extended_ppi_in_15[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_150[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150
extended_ppi_in_151[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151
extended_ppi_in_152[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152
extended_ppi_in_153[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153
extended_ppi_in_154[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154
extended_ppi_in_155[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155
extended_ppi_in_156[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156
extended_ppi_in_157[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157

Name	Protocol	Type	Description
extended_ppi_in_158[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158
extended_ppi_in_159[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159
extended_ppi_in_16[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_160[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160
extended_ppi_in_161[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161
extended_ppi_in_162[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162
extended_ppi_in_163[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163
extended_ppi_in_164[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164
extended_ppi_in_165[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165
extended_ppi_in_166[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166
extended_ppi_in_167[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167
extended_ppi_in_168[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168
extended_ppi_in_169[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169
extended_ppi_in_17[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_170[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170
extended_ppi_in_171[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171
extended_ppi_in_172[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172
extended_ppi_in_173[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173
extended_ppi_in_174[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174
extended_ppi_in_175[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175
extended_ppi_in_176[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176
extended_ppi_in_177[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177
extended_ppi_in_178[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178

Name	Protocol	Type	Description
extended_ppi_in_179[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179
extended_ppi_in_18[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_180[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180
extended_ppi_in_181[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181
extended_ppi_in_182[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182
extended_ppi_in_183[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183
extended_ppi_in_184[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184
extended_ppi_in_185[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185
extended_ppi_in_186[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186
extended_ppi_in_187[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187
extended_ppi_in_188[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188
extended_ppi_in_189[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189
extended_ppi_in_19[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.
extended_ppi_in_190[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190
extended_ppi_in_191[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191
extended_ppi_in_192[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192
extended_ppi_in_193[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193
extended_ppi_in_194[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194
extended_ppi_in_195[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195
extended_ppi_in_196[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196
extended_ppi_in_197[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197
extended_ppi_in_198[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198
extended_ppi_in_199[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199

Name	Protocol	Type	Description
extended_ppi_in_2[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_20[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_200[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200
extended_ppi_in_201[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201
extended_ppi_in_202[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202
extended_ppi_in_203[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203
extended_ppi_in_204[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204
extended_ppi_in_205[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205
extended_ppi_in_206[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206
extended_ppi_in_207[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207
extended_ppi_in_208[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208
extended_ppi_in_209[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209
extended_ppi_in_21[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.
extended_ppi_in_210[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210
extended_ppi_in_211[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211
extended_ppi_in_212[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212
extended_ppi_in_213[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213
extended_ppi_in_214[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214
extended_ppi_in_215[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215
extended_ppi_in_216[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216
extended_ppi_in_217[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217
extended_ppi_in_218[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218
extended_ppi_in_219[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219

Name	Protocol	Type	Description
extended_ppi_in_22[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.
extended_ppi_in_220[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220
extended_ppi_in_221[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221
extended_ppi_in_222[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222
extended_ppi_in_223[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223
extended_ppi_in_224[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224
extended_ppi_in_225[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225
extended_ppi_in_226[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226
extended_ppi_in_227[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227
extended_ppi_in_228[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228
extended_ppi_in_229[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229
extended_ppi_in_23[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_230[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230
extended_ppi_in_231[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231
extended_ppi_in_232[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232
extended_ppi_in_233[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233
extended_ppi_in_234[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234
extended_ppi_in_235[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235
extended_ppi_in_236[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236
extended_ppi_in_237[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237
extended_ppi_in_238[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238
extended_ppi_in_239[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239
extended_ppi_in_24[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.

Name	Protocol	Type	Description
extended_ppi_in_240[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240
extended_ppi_in_241[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241
extended_ppi_in_242[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242
extended_ppi_in_243[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243
extended_ppi_in_244[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244
extended_ppi_in_245[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245
extended_ppi_in_246[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246
extended_ppi_in_247[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247
extended_ppi_in_248[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248
extended_ppi_in_249[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249
extended_ppi_in_25[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_250[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250
extended_ppi_in_251[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251
extended_ppi_in_252[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252
extended_ppi_in_253[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253
extended_ppi_in_254[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254
extended_ppi_in_255[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255
extended_ppi_in_26[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.
extended_ppi_in_28[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_3[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.
extended_ppi_in_30[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.

Name	Protocol	Type	Description
extended_ppi_in_31[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.
extended_ppi_in_32[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.
extended_ppi_in_35[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.
extended_ppi_in_36[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_4[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_40[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.
extended_ppi_in_42[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.
extended_ppi_in_43[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.
extended_ppi_in_47[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.
extended_ppi_in_49[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_5[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_50[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.
extended_ppi_in_51[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.

Name	Protocol	Type	Description
extended_ppi_in_52[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.
extended_ppi_in_54[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.
extended_ppi_in_56[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_6[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_60[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.
extended_ppi_in_64[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64
extended_ppi_in_65[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65
extended_ppi_in_66[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66
extended_ppi_in_67[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67
extended_ppi_in_68[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68
extended_ppi_in_69[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69
extended_ppi_in_7[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_70[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70
extended_ppi_in_71[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71
extended_ppi_in_72[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72



Name	Protocol	Type	Description
extended_ppi_in_73[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73
extended_ppi_in_74[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74
extended_ppi_in_75[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75
extended_ppi_in_76[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76
extended_ppi_in_77[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77
extended_ppi_in_78[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78
extended_ppi_in_79[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79
extended_ppi_in_8[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.
extended_ppi_in_80[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80
extended_ppi_in_81[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81
extended_ppi_in_82[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82
extended_ppi_in_83[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83
extended_ppi_in_84[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84
extended_ppi_in_85[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85
extended_ppi_in_86[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86
extended_ppi_in_87[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87
extended_ppi_in_88[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88
extended_ppi_in_89[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89
extended_ppi_in_9[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.
extended_ppi_in_90[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90
extended_ppi_in_91[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91
extended_ppi_in_92[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92
extended_ppi_in_93[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93

Name	Protocol	Type	Description
extended_ppi_in_94[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94
extended_ppi_in_95[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95
extended_ppi_in_96[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96
extended_ppi_in_97[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97
extended_ppi_in_98[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98
extended_ppi_in_99[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99
extended_spi_in[1024]	Signal	Slave	Extended Shared peripheral interrupts.
icdrdt_out	PVBus	Master	AXI Stream Interface. This is used only when local cross chip addressing is enabled at the moment
icrdt_in	PVBus	Slave	-
po_reset	Signal	Slave	Reset.
ppi_in_0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_10[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_100[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 100
ppi_in_101[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 101
ppi_in_102[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 102
ppi_in_103[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 103
ppi_in_104[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 104
ppi_in_105[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 105
ppi_in_106[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 106
ppi_in_107[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 107
ppi_in_108[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 108
ppi_in_109[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 109
ppi_in_11[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_110[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 110
ppi_in_111[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 111
ppi_in_112[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 112
ppi_in_113[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 113
ppi_in_114[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 114
ppi_in_115[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 115
ppi_in_116[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 116
ppi_in_117[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 117
ppi_in_118[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 118
ppi_in_119[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 119

Name	Protocol	Type	Description
ppi_in_12[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_120[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 120
ppi_in_121[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 121
ppi_in_122[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 122
ppi_in_123[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 123
ppi_in_124[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 124
ppi_in_125[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 125
ppi_in_126[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 126
ppi_in_127[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 127
ppi_in_128[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 128
ppi_in_129[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 129
ppi_in_13[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_130[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 130
ppi_in_131[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 131
ppi_in_132[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 132
ppi_in_133[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 133
ppi_in_134[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 134
ppi_in_135[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 135
ppi_in_136[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 136
ppi_in_137[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 137
ppi_in_138[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 138
ppi_in_139[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 139
ppi_in_14[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_140[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 140
ppi_in_141[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 141
ppi_in_142[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 142
ppi_in_143[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 143
ppi_in_144[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 144
ppi_in_145[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 145
ppi_in_146[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 146
ppi_in_147[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 147
ppi_in_148[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 148
ppi_in_149[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 149
ppi_in_15[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_150[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 150
ppi_in_151[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 151
ppi_in_152[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 152
ppi_in_153[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 153
ppi_in_154[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 154

Name	Protocol	Type	Description
ppi_in_155[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 155
ppi_in_156[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 156
ppi_in_157[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 157
ppi_in_158[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 158
ppi_in_159[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 159
ppi_in_16[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_160[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 160
ppi_in_161[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 161
ppi_in_162[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 162
ppi_in_163[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 163
ppi_in_164[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 164
ppi_in_165[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 165
ppi_in_166[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 166
ppi_in_167[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 167
ppi_in_168[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 168
ppi_in_169[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 169
ppi_in_17[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_170[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 170
ppi_in_171[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 171
ppi_in_172[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 172
ppi_in_173[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 173
ppi_in_174[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 174
ppi_in_175[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 175
ppi_in_176[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 176
ppi_in_177[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 177
ppi_in_178[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 178
ppi_in_179[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 179
ppi_in_18[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_180[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 180
ppi_in_181[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 181
ppi_in_182[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 182
ppi_in_183[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 183
ppi_in_184[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 184
ppi_in_185[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 185
ppi_in_186[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 186
ppi_in_187[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 187
ppi_in_188[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 188
ppi_in_189[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 189
ppi_in_19[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 19.

Name	Protocol	Type	Description
ppi_in_190[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 190
ppi_in_191[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 191
ppi_in_192[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 192
ppi_in_193[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 193
ppi_in_194[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 194
ppi_in_195[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 195
ppi_in_196[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 196
ppi_in_197[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 197
ppi_in_198[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 198
ppi_in_199[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 199
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_20[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_200[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 200
ppi_in_201[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 201
ppi_in_202[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 202
ppi_in_203[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 203
ppi_in_204[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 204
ppi_in_205[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 205
ppi_in_206[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 206
ppi_in_207[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 207
ppi_in_208[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 208
ppi_in_209[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 209
ppi_in_21[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_210[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 210
ppi_in_211[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 211
ppi_in_212[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 212
ppi_in_213[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 213
ppi_in_214[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 214
ppi_in_215[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 215
ppi_in_216[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 216
ppi_in_217[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 217
ppi_in_218[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 218
ppi_in_219[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 219
ppi_in_22[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_220[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 220
ppi_in_221[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 221
ppi_in_222[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 222
ppi_in_223[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 223
ppi_in_224[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 224

Name	Protocol	Type	Description
ppi_in_225[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 225
ppi_in_226[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 226
ppi_in_227[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 227
ppi_in_228[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 228
ppi_in_229[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 229
ppi_in_23[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_230[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 230
ppi_in_231[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 231
ppi_in_232[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 232
ppi_in_233[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 233
ppi_in_234[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 234
ppi_in_235[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 235
ppi_in_236[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 236
ppi_in_237[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 237
ppi_in_238[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 238
ppi_in_239[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 239
ppi_in_24[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_240[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 240
ppi_in_241[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 241
ppi_in_242[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 242
ppi_in_243[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 243
ppi_in_244[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 244
ppi_in_245[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 245
ppi_in_246[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 246
ppi_in_247[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 247
ppi_in_248[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 248
ppi_in_249[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 249
ppi_in_25[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_250[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 250
ppi_in_251[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 251
ppi_in_252[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 252
ppi_in_253[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 253
ppi_in_254[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 254
ppi_in_255[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 255
ppi_in_26[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.

Name	Protocol	Type	Description
ppi_in_30[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_40[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_50[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_60[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 65.



Name	Protocol	Type	Description
ppi_in_66[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 66
ppi_in_67[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 67
ppi_in_68[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 68
ppi_in_69[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 69
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_70[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 70
ppi_in_71[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 71
ppi_in_72[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 72
ppi_in_73[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 73
ppi_in_74[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 74
ppi_in_75[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 75
ppi_in_76[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 76
ppi_in_77[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 77
ppi_in_78[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 78
ppi_in_79[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 79
ppi_in_8[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_80[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 80
ppi_in_81[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 81
ppi_in_82[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 82
ppi_in_83[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 83
ppi_in_84[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 84
ppi_in_85[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 85
ppi_in_86[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 86
ppi_in_87[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 87
ppi_in_88[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 88
ppi_in_89[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 89
ppi_in_9[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 9.
ppi_in_90[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 90
ppi_in_91[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 91
ppi_in_92[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 92
ppi_in_93[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 93
ppi_in_94[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 94
ppi_in_95[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 95
ppi_in_96[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 96
ppi_in_97[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 97
ppi_in_98[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 98
ppi_in_99[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 99
pvbus_filtermiss_m	PVBus	Master	Memory bus out. Transactions not filtered by the component.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.



Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Reset.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.

## Parameters for GIC700\_Filter

### CPU-affinities

#### Type

string

#### Default value

0.0.0.0

#### Description

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

### CPU-affinities-file

#### Type

string

#### Default value

""

#### Description

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

### GICD\_CTLR\_DS-1-means-secure-only

#### Type

bool

#### Default value

0x0

#### Description

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

### GICD\_TYPER2

#### Type

int

**Default value**

0x0

**Description**

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

**IIDR****Type**

int

**Default value**

0x403043b

**Description**

GICD\_IIDR and GICR\_IIDR value.

**ITS-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of interrupt bits supported by ITS.

**ITS-collection-ID-bits****Type**

int

**Default value**

0x8

**Description**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

**ITS-count****Type**

int

**Default value**

0x1

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**ITS-device-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits supported for ITS device IDs.

**ITS-enable-itt-address-verification****Type**

bool

**Default value**

0x0

**Description**

If true, a transaction will be sent to ITT Address for verification.

**ITS-hardware-collection-count****Type**

int

**Default value**

0x0

**Description**

Number of hardware collections held exclusively in the ITS.

**ITS-shared-vPE-table****Type**

int

**Default value**

0x0

**Description**

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

**ITS-vmovp-bit****Type**

bool

**Default value**

0x0

**Description**

Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

**PPI-count****Type**

int

**Default value**

16

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31

**SPI-blocks****Type**

int

**Default value**

0x3e

**Description**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

**add-output-cpu-wake-request-signal-from-redistributor****Type**

bool

**Default value**

0x0

**Description**

if true, the redistributor will have the output signal cpu\_wake\_request from GIC to DSU and if false, the signals are not added to the redistributor.

**affinity-width****Type**

string

**Default value**

"4.8.8.8"

**Description**

A dotted quad indicating the bitwidth of fields at each affinity level.

**allow-LPIEN-clear****Type**

bool

**Default value**

0x1

**Description**

Allow RW behaviour on GICR\_CTLR.LPIEN instead of set once.

**chip-count****Type**

int

**Default value**

0x10

**Description**

The total number of chips supported.

**chip-id****Type**

int

**Default value**

0x0

**Description**

Chip ID when multichip operation is enabled.

**chip-select-affinity-level****Type**

int

**Default value**

0x3

**Description**

Affinity level 2 or 3 can be used for chip select.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**common-vPE-table-affinity****Type**

string

**Default value**

""

**Description**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s) (where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on). Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when has-gicv4.1 is true.

**consolidators****Type**

string

**Default value**

""

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form:

```
'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1'
```

For example:

```
'0x3f100000:64:4096, 0x3f200000:64:4224'
```

The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

**cross-chip-AMBA-is-ACE****Type**

bool

**Default value**

0x0

**Description**

Indicates the AMBA protocol that the cross-chip interface uses. If true, the cross-chip interface uses the ACE5-Lite protocol. Otherwise, it uses the AXI5-Stream protocol. This parameter is meaningful only if enable-multichip-operation is set.

**enable-local-cross-chip-addressing****Type**

bool

**Default value**

0x0

**Description**

If true, each distributor in a multichip system will use routing table address (ADDR) values programmed by local software, and will not be overwritten by the default chip. Otherwise, all the distributors will share the same routing table address values programmed in the default chip. This parameter is valid only if enable-multichip-operation is set.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**extended-ppi-count****Type**

int

**Default value**

0x40

**Description**

Number of extended PPI supported.

**gicp-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

**gict-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

**has-gicv4.1****Type**

bool

**Default value**

0x1

**Description**

Enable GICv4.1 functionality; when false the component is inactive.

**has-two-security-states****Type**

bool

**Default value**

0x1

**Description**

If true, has two security states.

**has\_nmi****Type**

bool

**Default value**

0x0

**Description**

Enable support for Non-maskable Interrupts (NMIs). (FEAT\_GICv3\_NMI).

**max-cores-supported-by-GCI****Type**

int

**Default value**

0x8

**Description**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.



**max-pe-on-chip****Type**

int

**Default value**

0x4

**Description**

Maximum number of cores on any single chip. This will be used to identify the target chip and core.

**multichip-threaded-dgi****Type**

bool

**Default value**

0x1

**Description**

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if enable-multichip-operation is set.

**output\_attributes****Type**

string

**Default value**

"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID,  
ExtendedID[38]=MPAM\_NS"

**Description**

User-defined transform to be applied to bus attributes like MasterID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**prog-mpidr****Type**

unsigned

**Default value**

0

Whether software or hardware can remove cores from a GIC configuration.

**0**

none

**1**

prog - Secure software to remove cores during the boot up of a system.

**2**

strap - enables hardware to remove cores as GIC exits reset

**redistributor-group****Type**

string

**Default value**

-

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

**redistributor-group-file****Type**

string

**Default value**

-

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

**redistributor-power-managed-by-pwrr****Type**

bool

**Default value**

0x1

**Description**

GIC-700 dedistributor power management is done by updating GICR\_PWRR register.

**reg-base****Type**

int

**Default value**

0x2c010000

**Description**

GIC-700 base address.

### 3.10.42 GIC720AE

GIC-720AE IRI implementation: Single/multi chip validation/filter component variant limited to 64 PE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1012: IP revisions support**

Revision	Quality level
N/A	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Changes in 11.29.19**

Parameters added:

- `has_nmi`
- `view-id-bits-offset`

**About GIC720AE**

The model has the same functionality as [GIC700](#), but in addition supports the following AE-specific features:

- GIC FMU
- Multiple views

It has the same limitations as GIC700.

**Iris and MTI instances for GIC720AE**

This model has the following Iris instances:

**Table 3-1013: GIC720AE Iris instances**

InstanceName	ComponentName
GIC720AE	GIC_IRI
GIC720AE.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC720AE.ITS0	GICv4InterruptTranslationService
GIC720AE.ITS0.bus_subordinate	PVBusSlave
GIC720AE.fmu	FMU
GIC720AE.fmu.pvbus_slave	PVBusSlave
GIC720AE.rd_0	GICv4RedistributorInternal
GIC720AE.rd_0_0	GICv4RedistributorInternal
GIC720AE.rd_0_0_0	GICv4RedistributorInternal
GIC720AE.rd_0_0_0_0	GICv3Redistributor
GIC720AE.rd_tl	GICv3Distributor

This model has the following MTI trace components:

**Table 3-1014: GIC720AE MTI instances**

InstanceName	ComponentName
GIC720AE.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC720AE.ITS0	GICv4InterruptTranslationService
GIC720AE.ITS0.bus_subordinate	PVBusSlave
GIC720AE.fmu	FMU
GIC720AE.fmu.pvbus_slave	PVBusSlave
GIC720AE.rd_0	GICv4RedistributorInternal
GIC720AE.rd_0_0	GICv4RedistributorInternal
GIC720AE.rd_0_0_0	GICv4RedistributorInternal
GIC720AE.rd_0_0_0_0	GICv3Redistributor
GIC720AE.rd_tl	GICv3Distributor

## Ports for GIC720AE

**Table 3-1015: Ports**

Name	Protocol	Type	Description
apb_bus	PVBus	Slave	FMU signals
axi_stream_msi_s[32]	PVBus	Slave	AXI Stream Interface ports for MSI (message-signalled interrupt) translation, in GIC-720AE. Typically the SMMU's TCU connects to this port for MSI.
chip_id	Value	Slave	chip_id port for multichip operation
cpu_active_s[256]	Signal	Slave	CPUActive pins.
cpu_wake_request[256]	Signal	Master	-
extended_ppi_in_0[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.

Name	Protocol	Type	Description
extended_ppi_in_1[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
extended_ppi_in_10[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
extended_ppi_in_100[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100
extended_ppi_in_101[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101
extended_ppi_in_102[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102
extended_ppi_in_103[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103
extended_ppi_in_104[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104
extended_ppi_in_105[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105
extended_ppi_in_106[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106
extended_ppi_in_107[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107
extended_ppi_in_108[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108
extended_ppi_in_109[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109
extended_ppi_in_11[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
extended_ppi_in_110[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110
extended_ppi_in_111[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111
extended_ppi_in_112[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112
extended_ppi_in_113[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113
extended_ppi_in_114[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114
extended_ppi_in_115[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115
extended_ppi_in_116[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116
extended_ppi_in_117[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117
extended_ppi_in_118[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118
extended_ppi_in_119[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119

Name	Protocol	Type	Description
extended_ppi_in_12[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_120[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120
extended_ppi_in_121[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121
extended_ppi_in_122[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122
extended_ppi_in_123[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123
extended_ppi_in_124[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124
extended_ppi_in_125[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125
extended_ppi_in_126[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126
extended_ppi_in_127[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127
extended_ppi_in_128[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128
extended_ppi_in_129[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129
extended_ppi_in_13[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.
extended_ppi_in_130[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130
extended_ppi_in_131[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131
extended_ppi_in_132[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132
extended_ppi_in_133[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133
extended_ppi_in_134[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134
extended_ppi_in_135[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135
extended_ppi_in_136[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136
extended_ppi_in_137[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137
extended_ppi_in_138[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138
extended_ppi_in_139[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139
extended_ppi_in_14[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.

Name	Protocol	Type	Description
extended_ppi_in_140[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140
extended_ppi_in_141[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141
extended_ppi_in_142[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142
extended_ppi_in_143[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143
extended_ppi_in_144[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144
extended_ppi_in_145[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145
extended_ppi_in_146[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146
extended_ppi_in_147[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147
extended_ppi_in_148[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148
extended_ppi_in_149[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149
extended_ppi_in_15[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_150[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150
extended_ppi_in_151[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151
extended_ppi_in_152[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152
extended_ppi_in_153[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153
extended_ppi_in_154[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154
extended_ppi_in_155[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155
extended_ppi_in_156[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156
extended_ppi_in_157[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157
extended_ppi_in_158[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158
extended_ppi_in_159[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159
extended_ppi_in_16[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_160[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160

Name	Protocol	Type	Description
extended_ppi_in_161[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161
extended_ppi_in_162[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162
extended_ppi_in_163[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163
extended_ppi_in_164[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164
extended_ppi_in_165[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165
extended_ppi_in_166[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166
extended_ppi_in_167[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167
extended_ppi_in_168[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168
extended_ppi_in_169[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169
extended_ppi_in_17[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_170[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170
extended_ppi_in_171[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171
extended_ppi_in_172[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172
extended_ppi_in_173[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173
extended_ppi_in_174[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174
extended_ppi_in_175[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175
extended_ppi_in_176[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176
extended_ppi_in_177[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177
extended_ppi_in_178[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178
extended_ppi_in_179[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179
extended_ppi_in_18[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_180[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180
extended_ppi_in_181[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181



Name	Protocol	Type	Description
extended_ppi_in_182[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182
extended_ppi_in_183[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183
extended_ppi_in_184[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184
extended_ppi_in_185[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185
extended_ppi_in_186[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186
extended_ppi_in_187[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187
extended_ppi_in_188[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188
extended_ppi_in_189[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189
extended_ppi_in_19[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.
extended_ppi_in_190[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190
extended_ppi_in_191[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191
extended_ppi_in_192[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192
extended_ppi_in_193[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193
extended_ppi_in_194[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194
extended_ppi_in_195[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195
extended_ppi_in_196[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196
extended_ppi_in_197[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197
extended_ppi_in_198[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198
extended_ppi_in_199[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199
extended_ppi_in_2[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_20[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_200[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200
extended_ppi_in_201[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201

Name	Protocol	Type	Description
extended_ppi_in_202[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202
extended_ppi_in_203[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203
extended_ppi_in_204[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204
extended_ppi_in_205[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205
extended_ppi_in_206[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206
extended_ppi_in_207[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207
extended_ppi_in_208[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208
extended_ppi_in_209[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209
extended_ppi_in_21[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.
extended_ppi_in_210[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210
extended_ppi_in_211[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211
extended_ppi_in_212[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212
extended_ppi_in_213[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213
extended_ppi_in_214[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214
extended_ppi_in_215[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215
extended_ppi_in_216[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216
extended_ppi_in_217[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217
extended_ppi_in_218[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218
extended_ppi_in_219[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219
extended_ppi_in_22[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.
extended_ppi_in_220[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220
extended_ppi_in_221[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221
extended_ppi_in_222[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222

Name	Protocol	Type	Description
extended_ppi_in_223[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223
extended_ppi_in_224[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224
extended_ppi_in_225[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225
extended_ppi_in_226[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226
extended_ppi_in_227[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227
extended_ppi_in_228[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228
extended_ppi_in_229[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229
extended_ppi_in_23[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_230[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230
extended_ppi_in_231[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231
extended_ppi_in_232[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232
extended_ppi_in_233[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233
extended_ppi_in_234[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234
extended_ppi_in_235[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235
extended_ppi_in_236[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236
extended_ppi_in_237[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237
extended_ppi_in_238[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238
extended_ppi_in_239[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239
extended_ppi_in_24[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.
extended_ppi_in_240[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240
extended_ppi_in_241[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241
extended_ppi_in_242[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242
extended_ppi_in_243[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243

Name	Protocol	Type	Description
extended_ppi_in_244[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244
extended_ppi_in_245[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245
extended_ppi_in_246[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246
extended_ppi_in_247[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247
extended_ppi_in_248[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248
extended_ppi_in_249[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249
extended_ppi_in_25[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_250[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250
extended_ppi_in_251[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251
extended_ppi_in_252[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252
extended_ppi_in_253[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253
extended_ppi_in_254[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254
extended_ppi_in_255[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255
extended_ppi_in_26[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.
extended_ppi_in_28[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_3[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.
extended_ppi_in_30[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.
extended_ppi_in_31[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.
extended_ppi_in_32[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.

Name	Protocol	Type	Description
extended_ppi_in_35[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.
extended_ppi_in_36[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_4[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_40[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.
extended_ppi_in_42[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.
extended_ppi_in_43[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.
extended_ppi_in_47[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.
extended_ppi_in_49[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_5[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_50[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.
extended_ppi_in_51[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.
extended_ppi_in_52[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.
extended_ppi_in_54[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.

Name	Protocol	Type	Description
extended_ppi_in_56[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_6[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_60[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.
extended_ppi_in_64[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64
extended_ppi_in_65[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65
extended_ppi_in_66[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66
extended_ppi_in_67[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67
extended_ppi_in_68[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68
extended_ppi_in_69[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69
extended_ppi_in_7[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_70[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70
extended_ppi_in_71[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71
extended_ppi_in_72[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72
extended_ppi_in_73[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73
extended_ppi_in_74[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74
extended_ppi_in_75[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75
extended_ppi_in_76[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76

Name	Protocol	Type	Description
extended_ppi_in_77[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77
extended_ppi_in_78[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78
extended_ppi_in_79[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79
extended_ppi_in_8[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.
extended_ppi_in_80[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80
extended_ppi_in_81[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81
extended_ppi_in_82[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82
extended_ppi_in_83[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83
extended_ppi_in_84[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84
extended_ppi_in_85[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85
extended_ppi_in_86[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86
extended_ppi_in_87[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87
extended_ppi_in_88[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88
extended_ppi_in_89[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89
extended_ppi_in_9[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.
extended_ppi_in_90[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90
extended_ppi_in_91[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91
extended_ppi_in_92[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92
extended_ppi_in_93[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93
extended_ppi_in_94[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94
extended_ppi_in_95[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95
extended_ppi_in_96[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96
extended_ppi_in_97[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97

Name	Protocol	Type	Description
extended_ppi_in_98[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98
extended_ppi_in_99[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99
extended_spi_in[1024]	Signal	Slave	Extended Shared peripheral interrupts.
fmu_cri	Signal	Master	Critical Interrupt
fmu_eri	Signal	Master	Error recovery Interrupt
icdrt_out	PVBus	Master	AXI Stream Interface. This is used only when local cross chip addressing is enabled at the moment
icrdt_in	PVBus	Slave	-
po_reset	Signal	Slave	Resets.
ppi_in_0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_10[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_100[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 100
ppi_in_101[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 101
ppi_in_102[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 102
ppi_in_103[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 103
ppi_in_104[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 104
ppi_in_105[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 105
ppi_in_106[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 106
ppi_in_107[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 107
ppi_in_108[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 108
ppi_in_109[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 109
ppi_in_11[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_110[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 110
ppi_in_111[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 111
ppi_in_112[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 112
ppi_in_113[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 113
ppi_in_114[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 114
ppi_in_115[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 115
ppi_in_116[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 116
ppi_in_117[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 117
ppi_in_118[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 118
ppi_in_119[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 119
ppi_in_12[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_120[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 120
ppi_in_121[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 121
ppi_in_122[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 122



Name	Protocol	Type	Description
ppi_in_123[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 123
ppi_in_124[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 124
ppi_in_125[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 125
ppi_in_126[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 126
ppi_in_127[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 127
ppi_in_128[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 128
ppi_in_129[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 129
ppi_in_13[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_130[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 130
ppi_in_131[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 131
ppi_in_132[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 132
ppi_in_133[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 133
ppi_in_134[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 134
ppi_in_135[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 135
ppi_in_136[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 136
ppi_in_137[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 137
ppi_in_138[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 138
ppi_in_139[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 139
ppi_in_14[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_140[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 140
ppi_in_141[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 141
ppi_in_142[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 142
ppi_in_143[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 143
ppi_in_144[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 144
ppi_in_145[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 145
ppi_in_146[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 146
ppi_in_147[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 147
ppi_in_148[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 148
ppi_in_149[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 149
ppi_in_15[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_150[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 150
ppi_in_151[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 151
ppi_in_152[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 152
ppi_in_153[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 153
ppi_in_154[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 154
ppi_in_155[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 155
ppi_in_156[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 156
ppi_in_157[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 157
ppi_in_158[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 158

Name	Protocol	Type	Description
ppi_in_159[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 159
ppi_in_16[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_160[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 160
ppi_in_161[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 161
ppi_in_162[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 162
ppi_in_163[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 163
ppi_in_164[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 164
ppi_in_165[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 165
ppi_in_166[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 166
ppi_in_167[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 167
ppi_in_168[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 168
ppi_in_169[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 169
ppi_in_17[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_170[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 170
ppi_in_171[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 171
ppi_in_172[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 172
ppi_in_173[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 173
ppi_in_174[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 174
ppi_in_175[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 175
ppi_in_176[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 176
ppi_in_177[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 177
ppi_in_178[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 178
ppi_in_179[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 179
ppi_in_18[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_180[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 180
ppi_in_181[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 181
ppi_in_182[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 182
ppi_in_183[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 183
ppi_in_184[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 184
ppi_in_185[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 185
ppi_in_186[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 186
ppi_in_187[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 187
ppi_in_188[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 188
ppi_in_189[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 189
ppi_in_19[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_190[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 190
ppi_in_191[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 191
ppi_in_192[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 192
ppi_in_193[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 193

Name	Protocol	Type	Description
ppi_in_194[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 194
ppi_in_195[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 195
ppi_in_196[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 196
ppi_in_197[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 197
ppi_in_198[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 198
ppi_in_199[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 199
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_20[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_200[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 200
ppi_in_201[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 201
ppi_in_202[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 202
ppi_in_203[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 203
ppi_in_204[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 204
ppi_in_205[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 205
ppi_in_206[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 206
ppi_in_207[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 207
ppi_in_208[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 208
ppi_in_209[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 209
ppi_in_21[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_210[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 210
ppi_in_211[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 211
ppi_in_212[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 212
ppi_in_213[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 213
ppi_in_214[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 214
ppi_in_215[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 215
ppi_in_216[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 216
ppi_in_217[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 217
ppi_in_218[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 218
ppi_in_219[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 219
ppi_in_22[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_220[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 220
ppi_in_221[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 221
ppi_in_222[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 222
ppi_in_223[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 223
ppi_in_224[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 224
ppi_in_225[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 225
ppi_in_226[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 226
ppi_in_227[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 227
ppi_in_228[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 228

Name	Protocol	Type	Description
ppi_in_229[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 229
ppi_in_23[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_230[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 230
ppi_in_231[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 231
ppi_in_232[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 232
ppi_in_233[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 233
ppi_in_234[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 234
ppi_in_235[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 235
ppi_in_236[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 236
ppi_in_237[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 237
ppi_in_238[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 238
ppi_in_239[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 239
ppi_in_24[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_240[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 240
ppi_in_241[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 241
ppi_in_242[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 242
ppi_in_243[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 243
ppi_in_244[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 244
ppi_in_245[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 245
ppi_in_246[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 246
ppi_in_247[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 247
ppi_in_248[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 248
ppi_in_249[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 249
ppi_in_25[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_250[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 250
ppi_in_251[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 251
ppi_in_252[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 252
ppi_in_253[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 253
ppi_in_254[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 254
ppi_in_255[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 255
ppi_in_26[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_30[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 33.

Name	Protocol	Type	Description
ppi_in_34[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_40[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_50[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_60[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 69.

Name	Protocol	Type	Description
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_70[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 70
ppi_in_71[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 71
ppi_in_72[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 72
ppi_in_73[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 73
ppi_in_74[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 74
ppi_in_75[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 75
ppi_in_76[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 76
ppi_in_77[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 77
ppi_in_78[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 78
ppi_in_79[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 79
ppi_in_8[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_80[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 80
ppi_in_81[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 81
ppi_in_82[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 82
ppi_in_83[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 83
ppi_in_84[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 84
ppi_in_85[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 85
ppi_in_86[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 86
ppi_in_87[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 87
ppi_in_88[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 88
ppi_in_89[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 89
ppi_in_9[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 9.
ppi_in_90[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 90
ppi_in_91[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 91
ppi_in_92[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 92
ppi_in_93[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 93
ppi_in_94[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 94
ppi_in_95[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 95
ppi_in_96[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 96
ppi_in_97[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 97
ppi_in_98[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 98
ppi_in_99[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 99
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Resets.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.

## Parameters for GIC720AE

### **CPU-affinities**

**Type**

string

**Default value**

0.0.0.0

**Description**

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

### **CPU-affinities-file**

**Type**

string

**Default value**

""

**Description**

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

### **GICD\_CTLR-DS-1-means-secure-only**

**Type**

bool

**Default value**

0x0

**Description**

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

### **GICD\_TYPER2**

**Type**

int

**Default value**

0x0

**Description**

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

**IIDR****Type**

int

**Default value**

0x700143b

**Description**

GICD\_IIDR and GICR\_IIDR value.

**ITS-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of interrupt bits supported by ITS.

**ITS-collection-ID-bits****Type**

int

**Default value**

0x8

**Description**

Number of collection bits supported by ITS (optional parameter, 0 =&gt; 16bits support and GITS\_TYPER.CIL=0).

**ITS-count****Type**

int

**Default value**

0x1

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**ITS-device-bits****Type**

int

**Default value**

0x10



**Description**

Number of bits supported for ITS device IDs.

**ITS-enable-itt-address-verification****Type**

bool

**Default value**

0x0

**Description**

If true, a transaction will be sent to ITT Address for verification.

**ITS-hardware-collection-count****Type**

int

**Default value**

0x0

**Description**

Number of hardware collections held exclusively in the ITS.

**ITS-shared-vPE-table****Type**

int

**Default value**

0x0

**Description**

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

**ITS-vmovp-bit****Type**

bool

**Default value**

0x0

**Description**

Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

**PPI-count****Type**

int

**Default value**

16

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31

**SPI-blocks****Type**

int

**Default value**

0x3e

**Description**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

**add-output-cpu-wake-request-signal-from-redistributor****Type**

bool

**Default value**

0x0

**Description**

if true, the redistributor will have the output signal `cpu_wake_request` from GIC to DSU and if false, the signals are not added to the redistributor.

**affinity-width****Type**

string

**Default value**

"4.8.8.8"

**Description**

A dotted quad indicating the bitwidth of fields at each affinity level.

**allow-LPIEN-clear****Type**

bool

**Default value**

0x1

**Description**

Allow RW behaviour on GICR\_CTLR.LPIEN instead of set once.

**chip-count****Type**

int

**Default value**

0x10

**Description**

The total number of chips supported.

**chip-id****Type**

int

**Default value**

0x0

**Description**

Chip ID when multichip operation is enabled.

**chip-select-affinity-level****Type**

int

**Default value**

0x3

**Description**

Affinity level 2 or 3 can be used for chip select.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**common-vPE-table-affinity****Type**

string

**Default value**

""

**Description**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s) (where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on). Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when has-gicv4.1 is true.

**consolidators****Type**

string

**Default value**

""

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form:

```
'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1'
```

For example:

```
'0x3f100000:64:4096, 0x3f200000:64:4224'
```

The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

**cross-chip-AMBA-is-ACE****Type**

bool

**Default value**

0x0

**Description**

Indicates the AMBA protocol that the cross-chip interface uses. If true, the cross-chip interface uses the ACE5-Lite protocol. Otherwise, it uses the AXI5-Stream protocol. This parameter is meaningful only if enable-multichip-operation is set.

**enable-local-cross-chip-addressing****Type**

bool

**Default value**

0x0

**Description**

If true, each distributor in a multichip system will use routing table address (ADDR) values programmed by local software, and will not be overwritten by the default chip. Otherwise, all the distributors will share the same routing table address values programmed in the default chip. This parameter is valid only if enable-multichip-operation is set.

**enable-multiple-views-feature****Type**

bool

**Default value**

0x0

**Description**

If true, multiple view feature will provide multiple programming views which can be used by multiple hypervisors.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**extended-ppi-count****Type**

int

**Default value**

0x40

**Description**

Number of extended PPI supported.

**fmu-blktype-num****Type**

int

**Default value**

0x6

**Description**

Number of stakeholder block types for FMU.

**`gicp-allow-ns-reset`****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

**`gict-allow-ns-reset`****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

**`has-gicv4.1`****Type**

bool

**Default value**

0x1

**Description**

Enable GICv4.1 functionality; when false the component is inactive.

**`has-two-security-states`****Type**

bool

**Default value**

0x1

**Description**

If true, has two security states.

**has\_nmi****Type**

bool

**Default value**

0x0

**Description**

Enable support for Non-maskable Interrupts (NMIs). (FEAT\_GICv3\_NMI).

**max-cores-supported-by-GCI****Type**

int

**Default value**

0x8

**Description**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

**max-pe-on-chip****Type**

int

**Default value**

0x4

**Description**

Maximum number of cores on any single chip. This will be used to identify the target chip and core.

**multichip-threaded-dgi****Type**

bool

**Default value**

0x1

**Description**

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if enable-multichip-operation is set.

**output\_attributes****Type**

string

**Default value**

"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID,  
ExtendedID[38]=MPAM\_NS"

**Description**

User-defined transform to be applied to bus attributes like MasterID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**prog-mpidr****Type**

unsigned

**Default value**

0

Whether software or hardware can remove cores from a GIC configuration.

0

none

1

prog - Secure software to remove cores during the boot up of a system.

2

strap - enables hardware to remove cores as GIC exits reset

**redistributor-group****Type**

string

**Default value**

-

Redistributor grouping information with affinity as JSON :

```
{
  "0":
    ["0.0.0.0",
     "0.0.0.1"],
  "1":
    ["0.0.1.0",
     "0.0.1.1"]
}
```



where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

### **redistributor-group-file**

#### **Type**

string

#### **Default value**

-

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

### **redistributor-power-managed-by-pwrr**

#### **Type**

bool

#### **Default value**

0x1

#### **Description**

GIC-700 dedistributor power management is done by updating GICR\_PWRR register.

### **reg-base**

#### **Type**

int

#### **Default value**

0x2c010000

#### **Description**

GIC-700 base address.

### **view-id-bits-offset**

#### **Type**

int

#### **Default value**

0x0

#### **Description**

The offset of the view-id bits in the address. Effective only when multiple views feature is enabled. Accepted values will depend on the configuration affecting the GIC memory footprint, such as the input to the parameter 'CPU-affinities'. For example, for a 32MB-sized view, the value of this parameter should be 25.

### 3.10.43 GIC720AE\_Filter

GIC-720AE IRI implementation: Single/multi chip validation/filter component variant limited to 64 PE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1016: IP revisions support**

Revision	Quality level
N/A	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Changes in 11.29.19

Parameters added:

- `has_nmi`
- `view-id-bits-offset`

#### About GIC720AE\_Filter

The model has the same functionality as [GIC700\\_Filter](#), but in addition supports the following AE-specific features:

- GIC FMU
- Multiple views

It has the same limitations as [GIC700\\_Filter](#).

#### Iris and MTI instances for GIC720AE\_Filter

This model has the following Iris instances:

**Table 3-1017: GIC720AE\_Filter Iris instances**

InstanceName	ComponentName
GIC720AE_Filter	GIC_IRI
GIC720AE_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC720AE_Filter.ITS0	GICv4InterruptTranslationService
GIC720AE_Filter.ITS0.bus_subordinate	PVBusSlave
GIC720AE_Filter.fmu	FMU
GIC720AE_Filter.fmu.pvbus_slave	PVBusSlave
GIC720AE_Filter.rd_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0_0_0	GICv3Redistributor
GIC720AE_Filter.rd_tl	GICv3Distributor

This model has the following MTI trace components:

**Table 3-1018: GIC720AE\_Filter MTI instances**

InstanceName	ComponentName
GIC720AE_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC720AE_Filter.ITS0	GICv4InterruptTranslationService
GIC720AE_Filter.ITS0.bus_subordinate	PVBusSlave
GIC720AE_Filter.fmu	FMU
GIC720AE_Filter.fmu.pvbus_slave	PVBusSlave
GIC720AE_Filter.rd_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0_0_0	GICv3Redistributor
GIC720AE_Filter.rd_tl	GICv3Distributor

### Ports for GIC720AE\_Filter

**Table 3-1019: Ports**

Name	Protocol	Type	Description
apb_bus	PVBus	Slave	FMU signals
axi_stream_msi_s[32]	PVBus	Slave	AXI Stream Interface ports for MSI (message-signalled interrupt) translation, in GIC-720AE. Typically the SMMU's TCU connects to this port for MSI.
chip_id	Value	Slave	chip_id port for multichip operation
cpu_active_s[256]	Signal	Slave	CPUActive pins.
extended_ppi_in_0[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
extended_ppi_in_1[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
extended_ppi_in_10[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
extended_ppi_in_100[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100
extended_ppi_in_101[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101
extended_ppi_in_102[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102
extended_ppi_in_103[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103
extended_ppi_in_104[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104
extended_ppi_in_105[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105
extended_ppi_in_106[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106

Name	Protocol	Type	Description
extended_ppi_in_107[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107
extended_ppi_in_108[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108
extended_ppi_in_109[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109
extended_ppi_in_11[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
extended_ppi_in_110[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110
extended_ppi_in_111[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111
extended_ppi_in_112[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112
extended_ppi_in_113[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113
extended_ppi_in_114[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114
extended_ppi_in_115[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115
extended_ppi_in_116[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116
extended_ppi_in_117[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117
extended_ppi_in_118[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118
extended_ppi_in_119[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119
extended_ppi_in_12[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_120[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120
extended_ppi_in_121[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121
extended_ppi_in_122[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122
extended_ppi_in_123[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123
extended_ppi_in_124[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124
extended_ppi_in_125[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125
extended_ppi_in_126[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126
extended_ppi_in_127[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127

Name	Protocol	Type	Description
extended_ppi_in_128[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128
extended_ppi_in_129[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129
extended_ppi_in_13[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.
extended_ppi_in_130[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130
extended_ppi_in_131[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131
extended_ppi_in_132[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132
extended_ppi_in_133[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133
extended_ppi_in_134[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134
extended_ppi_in_135[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135
extended_ppi_in_136[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136
extended_ppi_in_137[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137
extended_ppi_in_138[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138
extended_ppi_in_139[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139
extended_ppi_in_14[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.
extended_ppi_in_140[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140
extended_ppi_in_141[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141
extended_ppi_in_142[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142
extended_ppi_in_143[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143
extended_ppi_in_144[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144
extended_ppi_in_145[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145
extended_ppi_in_146[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146
extended_ppi_in_147[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147
extended_ppi_in_148[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148

Name	Protocol	Type	Description
extended_ppi_in_149[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149
extended_ppi_in_15[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_150[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150
extended_ppi_in_151[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151
extended_ppi_in_152[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152
extended_ppi_in_153[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153
extended_ppi_in_154[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154
extended_ppi_in_155[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155
extended_ppi_in_156[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156
extended_ppi_in_157[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157
extended_ppi_in_158[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158
extended_ppi_in_159[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159
extended_ppi_in_16[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_160[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160
extended_ppi_in_161[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161
extended_ppi_in_162[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162
extended_ppi_in_163[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163
extended_ppi_in_164[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164
extended_ppi_in_165[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165
extended_ppi_in_166[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166
extended_ppi_in_167[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167
extended_ppi_in_168[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168
extended_ppi_in_169[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169

Name	Protocol	Type	Description
extended_ppi_in_17[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_170[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170
extended_ppi_in_171[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171
extended_ppi_in_172[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172
extended_ppi_in_173[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173
extended_ppi_in_174[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174
extended_ppi_in_175[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175
extended_ppi_in_176[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176
extended_ppi_in_177[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177
extended_ppi_in_178[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178
extended_ppi_in_179[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179
extended_ppi_in_18[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_180[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180
extended_ppi_in_181[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181
extended_ppi_in_182[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182
extended_ppi_in_183[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183
extended_ppi_in_184[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184
extended_ppi_in_185[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185
extended_ppi_in_186[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186
extended_ppi_in_187[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187
extended_ppi_in_188[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188
extended_ppi_in_189[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189
extended_ppi_in_19[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.

Name	Protocol	Type	Description
extended_ppi_in_190[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190
extended_ppi_in_191[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191
extended_ppi_in_192[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192
extended_ppi_in_193[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193
extended_ppi_in_194[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194
extended_ppi_in_195[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195
extended_ppi_in_196[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196
extended_ppi_in_197[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197
extended_ppi_in_198[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198
extended_ppi_in_199[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199
extended_ppi_in_2[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_20[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_200[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200
extended_ppi_in_201[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201
extended_ppi_in_202[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202
extended_ppi_in_203[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203
extended_ppi_in_204[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204
extended_ppi_in_205[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205
extended_ppi_in_206[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206
extended_ppi_in_207[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207
extended_ppi_in_208[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208
extended_ppi_in_209[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209
extended_ppi_in_21[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.



Name	Protocol	Type	Description
extended_ppi_in_210[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210
extended_ppi_in_211[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211
extended_ppi_in_212[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212
extended_ppi_in_213[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213
extended_ppi_in_214[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214
extended_ppi_in_215[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215
extended_ppi_in_216[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216
extended_ppi_in_217[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217
extended_ppi_in_218[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218
extended_ppi_in_219[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219
extended_ppi_in_22[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.
extended_ppi_in_220[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220
extended_ppi_in_221[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221
extended_ppi_in_222[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222
extended_ppi_in_223[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223
extended_ppi_in_224[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224
extended_ppi_in_225[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225
extended_ppi_in_226[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226
extended_ppi_in_227[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227
extended_ppi_in_228[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228
extended_ppi_in_229[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229
extended_ppi_in_23[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_230[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230

Name	Protocol	Type	Description
extended_ppi_in_231[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231
extended_ppi_in_232[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232
extended_ppi_in_233[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233
extended_ppi_in_234[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234
extended_ppi_in_235[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235
extended_ppi_in_236[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236
extended_ppi_in_237[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237
extended_ppi_in_238[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238
extended_ppi_in_239[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239
extended_ppi_in_24[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.
extended_ppi_in_240[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240
extended_ppi_in_241[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241
extended_ppi_in_242[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242
extended_ppi_in_243[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243
extended_ppi_in_244[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244
extended_ppi_in_245[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245
extended_ppi_in_246[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246
extended_ppi_in_247[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247
extended_ppi_in_248[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248
extended_ppi_in_249[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249
extended_ppi_in_25[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_250[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250
extended_ppi_in_251[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251

Name	Protocol	Type	Description
extended_ppi_in_252[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252
extended_ppi_in_253[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253
extended_ppi_in_254[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254
extended_ppi_in_255[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255
extended_ppi_in_26[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.
extended_ppi_in_28[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_3[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.
extended_ppi_in_30[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.
extended_ppi_in_31[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.
extended_ppi_in_32[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.
extended_ppi_in_35[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.
extended_ppi_in_36[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_4[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_40[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.
extended_ppi_in_42[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.

Name	Protocol	Type	Description
extended_ppi_in_43[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.
extended_ppi_in_47[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.
extended_ppi_in_49[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_5[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_50[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.
extended_ppi_in_51[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.
extended_ppi_in_52[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.
extended_ppi_in_54[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.
extended_ppi_in_56[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_6[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_60[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.

Name	Protocol	Type	Description
extended_ppi_in_64[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64
extended_ppi_in_65[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65
extended_ppi_in_66[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66
extended_ppi_in_67[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67
extended_ppi_in_68[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68
extended_ppi_in_69[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69
extended_ppi_in_7[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_70[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70
extended_ppi_in_71[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71
extended_ppi_in_72[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72
extended_ppi_in_73[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73
extended_ppi_in_74[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74
extended_ppi_in_75[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75
extended_ppi_in_76[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76
extended_ppi_in_77[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77
extended_ppi_in_78[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78
extended_ppi_in_79[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79
extended_ppi_in_8[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.
extended_ppi_in_80[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80
extended_ppi_in_81[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81
extended_ppi_in_82[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82
extended_ppi_in_83[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83
extended_ppi_in_84[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84

Name	Protocol	Type	Description
extended_ppi_in_85[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85
extended_ppi_in_86[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86
extended_ppi_in_87[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87
extended_ppi_in_88[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88
extended_ppi_in_89[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89
extended_ppi_in_9[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.
extended_ppi_in_90[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90
extended_ppi_in_91[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91
extended_ppi_in_92[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92
extended_ppi_in_93[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93
extended_ppi_in_94[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94
extended_ppi_in_95[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95
extended_ppi_in_96[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96
extended_ppi_in_97[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97
extended_ppi_in_98[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98
extended_ppi_in_99[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99
extended_spi_in[1024]	Signal	Slave	Extended Shared peripheral interrupts.
fmu_cri	Signal	Master	Critical Interrupt
fmu_eri	Signal	Master	Error recovery Interrupt
icdrt_out	PVBus	Master	AXI Stream Interface. This is used only when local cross chip addressing is enabled at the moment
icrdt_in	PVBus	Slave	-
po_reset	Signal	Slave	Reset.
ppi_in_0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_10[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_100[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 100
ppi_in_101[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 101

Name	Protocol	Type	Description
ppi_in_102[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 102
ppi_in_103[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 103
ppi_in_104[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 104
ppi_in_105[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 105
ppi_in_106[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 106
ppi_in_107[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 107
ppi_in_108[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 108
ppi_in_109[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 109
ppi_in_11[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_110[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 110
ppi_in_111[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 111
ppi_in_112[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 112
ppi_in_113[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 113
ppi_in_114[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 114
ppi_in_115[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 115
ppi_in_116[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 116
ppi_in_117[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 117
ppi_in_118[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 118
ppi_in_119[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 119
ppi_in_12[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_120[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 120
ppi_in_121[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 121
ppi_in_122[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 122
ppi_in_123[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 123
ppi_in_124[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 124
ppi_in_125[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 125
ppi_in_126[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 126
ppi_in_127[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 127
ppi_in_128[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 128
ppi_in_129[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 129
ppi_in_13[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_130[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 130
ppi_in_131[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 131
ppi_in_132[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 132
ppi_in_133[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 133
ppi_in_134[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 134
ppi_in_135[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 135
ppi_in_136[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 136
ppi_in_137[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 137

Name	Protocol	Type	Description
ppi_in_138[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 138
ppi_in_139[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 139
ppi_in_14[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_140[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 140
ppi_in_141[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 141
ppi_in_142[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 142
ppi_in_143[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 143
ppi_in_144[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 144
ppi_in_145[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 145
ppi_in_146[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 146
ppi_in_147[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 147
ppi_in_148[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 148
ppi_in_149[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 149
ppi_in_15[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_150[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 150
ppi_in_151[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 151
ppi_in_152[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 152
ppi_in_153[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 153
ppi_in_154[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 154
ppi_in_155[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 155
ppi_in_156[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 156
ppi_in_157[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 157
ppi_in_158[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 158
ppi_in_159[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 159
ppi_in_16[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_160[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 160
ppi_in_161[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 161
ppi_in_162[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 162
ppi_in_163[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 163
ppi_in_164[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 164
ppi_in_165[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 165
ppi_in_166[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 166
ppi_in_167[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 167
ppi_in_168[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 168
ppi_in_169[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 169
ppi_in_17[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_170[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 170
ppi_in_171[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 171
ppi_in_172[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 172



Name	Protocol	Type	Description
ppi_in_173[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 173
ppi_in_174[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 174
ppi_in_175[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 175
ppi_in_176[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 176
ppi_in_177[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 177
ppi_in_178[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 178
ppi_in_179[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 179
ppi_in_18[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_180[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 180
ppi_in_181[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 181
ppi_in_182[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 182
ppi_in_183[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 183
ppi_in_184[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 184
ppi_in_185[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 185
ppi_in_186[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 186
ppi_in_187[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 187
ppi_in_188[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 188
ppi_in_189[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 189
ppi_in_19[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_190[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 190
ppi_in_191[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 191
ppi_in_192[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 192
ppi_in_193[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 193
ppi_in_194[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 194
ppi_in_195[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 195
ppi_in_196[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 196
ppi_in_197[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 197
ppi_in_198[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 198
ppi_in_199[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 199
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_20[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_200[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 200
ppi_in_201[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 201
ppi_in_202[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 202
ppi_in_203[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 203
ppi_in_204[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 204
ppi_in_205[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 205
ppi_in_206[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 206
ppi_in_207[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 207

Name	Protocol	Type	Description
ppi_in_208[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 208
ppi_in_209[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 209
ppi_in_21[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_210[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 210
ppi_in_211[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 211
ppi_in_212[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 212
ppi_in_213[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 213
ppi_in_214[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 214
ppi_in_215[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 215
ppi_in_216[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 216
ppi_in_217[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 217
ppi_in_218[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 218
ppi_in_219[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 219
ppi_in_22[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_220[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 220
ppi_in_221[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 221
ppi_in_222[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 222
ppi_in_223[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 223
ppi_in_224[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 224
ppi_in_225[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 225
ppi_in_226[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 226
ppi_in_227[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 227
ppi_in_228[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 228
ppi_in_229[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 229
ppi_in_23[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_230[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 230
ppi_in_231[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 231
ppi_in_232[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 232
ppi_in_233[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 233
ppi_in_234[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 234
ppi_in_235[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 235
ppi_in_236[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 236
ppi_in_237[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 237
ppi_in_238[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 238
ppi_in_239[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 239
ppi_in_24[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_240[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 240
ppi_in_241[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 241
ppi_in_242[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 242

Name	Protocol	Type	Description
ppi_in_243[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 243
ppi_in_244[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 244
ppi_in_245[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 245
ppi_in_246[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 246
ppi_in_247[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 247
ppi_in_248[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 248
ppi_in_249[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 249
ppi_in_25[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_250[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 250
ppi_in_251[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 251
ppi_in_252[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 252
ppi_in_253[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 253
ppi_in_254[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 254
ppi_in_255[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 255
ppi_in_26[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_30[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_40[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 48.

Name	Protocol	Type	Description
ppi_in_49[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_50[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_60[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_70[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_8[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_80[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 83.

Name	Protocol	Type	Description
ppi_in_84[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 84
ppi_in_85[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 85
ppi_in_86[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 86
ppi_in_87[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 87
ppi_in_88[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 88
ppi_in_89[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 89
ppi_in_9[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 9.
ppi_in_90[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 90
ppi_in_91[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 91
ppi_in_92[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 92
ppi_in_93[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 93
ppi_in_94[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 94
ppi_in_95[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 95
ppi_in_96[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 96
ppi_in_97[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 97
ppi_in_98[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 98
ppi_in_99[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 99
pvbus_filtermiss_m	PVBus	Master	Memory bus out. Transactions not filtered by the component.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Reset.
spi_in[960]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.

## Parameters for GIC720AE\_Filter

### CPU-affinities

#### Type

string

#### Default value

0.0.0.0

#### Description

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

### CPU-affinities-file

#### Type

string

**Default value**

""

**Description**

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

**GICD\_CTLR-DS-1-means-secure-only****Type**

bool

**Default value**

0x0

**Description**

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

**GICD\_TYPER2****Type**

int

**Default value**

0x0

**Description**

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

**IIDR****Type**

int

**Default value**

0x700143b

**Description**

GICD\_IIDR and GICR\_IIDR value.

**ITS-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of interrupt bits supported by ITS.

**ITS-collection-ID-bits****Type**

int

**Default value**

0x8

**Description**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

**ITS-count****Type**

int

**Default value**

0x1

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**ITS-device-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits supported for ITS device IDs.

**ITS-enable-itt-address-verification****Type**

bool

**Default value**

0x0

**Description**

If true, a transaction will be sent to ITT Address for verification.

**ITS-hardware-collection-count****Type**

int

**Default value**

0x0

**Description**  
Number of hardware collections held exclusively in the ITS.

**ITS-shared-vPE-table**

**Type**  
int  
**Default value**  
0x0

**Description**  
Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

**ITS-vmovp-bit**

**Type**  
bool  
**Default value**  
0x0

**Description**  
Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

**PPI-count**

**Type**  
int  
**Default value**  
16

Selects the number of PPI available for each PE

**8**  
id22-27,29,30

**12**  
id 20-31

**16**  
id 16-31

**SPI-blocks**

**Type**  
int  
**Default value**  
0x3e



**Description**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

**add-output-cpu-wake-request-signal-from-redistributor****Type**

bool

**Default value**

0x0

**Description**

if true, the redistributor will have the output signal cpu\_wake\_request from GIC to DSU and if false, the signals are not added to the redistributor.

**affinity-width****Type**

string

**Default value**

"4.8.8.8"

**Description**

A dotted quad indicating the bitwidth of fields at each affinity level.

**allow-LPIEN-clear****Type**

bool

**Default value**

0x1

**Description**

Allow RW behaviour on GICR\_CTLR.LPIEN instead of set once.

**chip-count****Type**

int

**Default value**

0x10

**Description**

The total number of chips supported.

**chip-id****Type**

int

**Default value**

0x0

**Description**

Chip ID when multichip operation is enabled.

**chip-select-affinity-level****Type**

int

**Default value**

0x3

**Description**

Affinity level 2 or 3 can be used for chip select.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**common-vPE-table-affinity****Type**

string

**Default value**

""

**Description**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s) (where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on). Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when has-gicv4.1 is true.

**consolidators****Type**

string

**Default value**

""

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form:

```
'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1'
```

For example:

```
'0x3f100000:64:4096, 0x3f200000:64:4224'
```

The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

### **cross-chip-AMBA-is-ACE**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

Indicates the AMBA protocol that the cross-chip interface uses. If true, the cross-chip interface uses the ACE5-Lite protocol. Otherwise, it uses the AXI5-Stream protocol. This parameter is meaningful only if enable-multichip-operation is set.

### **enable-local-cross-chip-addressing**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

If true, each distributor in a multichip system will use routing table address (ADDR) values programmed by local software, and will not be overwritten by the default chip. Otherwise, all the distributors will share the same routing table address values programmed in the default chip. This parameter is valid only if enable-multichip-operation is set.

### **enable-multiple-views-feature**

#### **Type**

bool

#### **Default value**

0x0

#### **Description**

If true, multiple view feature will provide multiple programming views which can be used by multiple hypervisors.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**extended-ppi-count****Type**

int

**Default value**

0x40

**Description**

Number of extended PPI supported.

**fmv-blktype-num****Type**

int

**Default value**

0x6

**Description**

Number of stakeholder block types for FMU.

**gicp-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

**gict-allow-ns-reset****Type**

bool

**Default value**

0x1

**Description**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

**has-gicv4.1****Type**

bool

**Default value**

0x1

**Description**

Enable GICv4.1 functionality; when false the component is inactive.

**has-two-security-states****Type**

bool

**Default value**

0x1

**Description**

If true, has two security states.

**has\_nmi****Type**

bool

**Default value**

0x0

**Description**

Enable support for Non-maskable Interrupts (NMIs). (FEAT\_GICv3\_NMI).

**max-cores-supported-by-GCI****Type**

int

**Default value**

0x8

**Description**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

**max-pe-on-chip****Type**

int

**Default value**

0x4

**Description**

Maximum number of cores on any single chip. This will be used to identify the target chip and core.

**multichip-threaded-dgi****Type**

bool

**Default value**

0x1

**Description**

Enable sending multichip DGI messages in a separate thread. This parameter is valid only if enable-multichip-operation is set.

**output\_attributes****Type**

string

**Default value**

"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID,  
ExtendedID[38]=MPAM\_NS"

**Description**

User-defined transform to be applied to bus attributes like MasterID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**prog-mpidr****Type**

unsigned

**Default value**

0

Whether software or hardware can remove cores from a GIC configuration.

**0**

none

**1**

prog - Secure software to remove cores during the boot up of a system.

**2**

strap - enables hardware to remove cores as GIC exits reset

**redistributor-group****Type**

string

**Default value**

-

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

**redistributor-group-file****Type**

string

**Default value**

-

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

**redistributor-power-managed-by-pwrr****Type**

bool

**Default value**

0x1

**Description**

GIC-700 dedistributor power management is done by updating GICR\_PWRR register.

**reg-base**

**Type**

int

**Default value**

0x2c010000

**Description**

GIC-700 base address.

**view-id-bits-offset**

**Type**

int

**Default value**

0x0

**Description**

The offset of the view-id bits in the address. Effective only when multiple views feature is enabled. Accepted values will depend on the configuration affecting the GIC memory footprint, such as the input to the parameter 'CPU-affinities'. For example, for a 32MB-sized view, the value of this parameter should be 25.

3.10.44 GIC\_400

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1020: IP revisions support

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About GIC\_400

This component is a wrapper that permits easier configuration of the v7\_VGIC component that supports parameterized configuration.

The GIC-400 has several memory-mapped interfaces at the same address. The processor that is communicating with the GIC-400 banks them. The GIC-400 must be able to identify which processor a transaction originates from. In the hardware, the AUSER fields on AXI supply this information to the GIC-400. In the model, there is no exact equivalent to this field. However, each transaction has a `master_id` that the model can use to identify the originating processor.

Arm clusters assign the `master_id` as follows:



**Bits[31:16]**

SBZ, which the GIC-400 ignores.

**Bits[5:2]**

CLUSTERID.

**Bits[1:0]**

cpu\_id within the cluster.

CLUSTERID is the 4-bit field that either a parameter on the processor sets or a value that the `clusterid` port drives. CPUID is the core number within the cluster. CLUSTERID appears in the CP15 register space as part of the MPIDR register.

The Arm architecture suggests that each cluster in the system is given a different CLUSTERID. This distinction is essential for the VGIC to identify the cluster. The parameters in the GIC-400 component permit it to construct the map of `master_id` to interface number.

Processor interfaces that the GIC-400 supports have these parameters:

- `interfaceN.cluster_id`
- `interfaceN.core_id`
- `interfaceN.inout_port_number_to_use`

N is the interface number (0-7). The `cluster_id` and `core_id` tell the GIC-400 to map that cluster or core combination to interface N.

By using `inout_port_number_to_use`, the GIC-400 has some input and output ports that pair with a particular processor interface. For example:

- The `irqcpu[]` pin wires to the `irq` port of the corresponding processor.
- The `cntpnsirq` pin from the processor wires to a `cntpnsirq[]` pin on GIC-400 to transport a Private Peripheral Interrupt (PPI) from the processor to the GIC-400.

The `interfaceN.inout_port_number_to_use` parameter supports clusters that can have variable numbers of cores. It tells the GIC-400 that to send to or receive a signal from the processor that is attached to interface N, it must use these pins:

- `irqout[interfaceN.inout_port_number_to_use]`
- `fiqout[interfaceN.inout_port_number_to_use]`
- `virqout[interfaceN.inout_port_number_to_use]`
- `vfiqout[interfaceN.inout_port_number_to_use]`
- `legacyirq[interfaceN.inout_port_number_to_use]`
- `cntpnsirq[interfaceN.inout_port_number_to_use]`
- `cntpsirq[interfaceN.inout_port_number_to_use]`
- `legacyfiq[interfaceN.inout_port_number_to_use]`
- `cntvirq[interfaceN.inout_port_number_to_use]`

- `cnthpirq[interfaceN.inout_port_number_to_use]`
- ...

`legacyirq` and `legacyfiq` are not signals from the processor but are signals into the GIC-400 from the legacy interrupt system. They are wired to PPIs. If the control registers of the GIC-400 are set up in particular ways, they can also bypass the GIC-400. See [ARM Generic Interrupt Controller Architecture version 2.0 Architecture Specification](#) for more information.

The fabric between the clusters and the GIC might remap the `master_id` of a transaction. If so, then the GIC might lose the ability to identify the originating processor. The fabrics that Arm ships in Fast Models perform no such transformation.

The comparison that the GIC-400 performs on the `master_id` is only on the bottom 6 bits of the `master_id`. It ignores the rest. If you are writing your own fabric and do not properly propagate the `master_id` or transform it, the GIC-400 might not be able to identify the processor. The source code for the GIC\_400 component can be examined to see how it might be adapted for it to understand different `master_id` schemes.

## Differences between the model and the RTL

The GIC-400 model has these limitations:

- Reads and writes to `GICD_ISACTIVERn`, `GICD_ICACTIVERn`, `GICD_ISPENDRn`, or `GICD_ICPENDRn` might not work as expected unless there is a configured target in `GICD_ICFGRm`.
- Some of the interaction between `GICD_CTLR.EnableGrpX` and level-sensitive interrupts might not work correctly.
- It does not model the `nIRQOUT` or `nFIQOUT` signals.
- It models interrupts with positive logic, rather than the negative logic that the hardware uses. So, the signal pins omit the 'n' prefix in their names.

## Iris and MTI instances for GIC\_400

This model has the following Iris instances:

**Table 3-1021: GIC\_400 Iris instances**

InstanceName	ComponentName
<code>GIC_400</code>	<code>GIC_400</code>
<code>GIC_400.vgic_bus_slave</code>	<code>PVBusSlave</code>

This model has the following MTI trace components:

**Table 3-1022: GIC\_400 MTI instances**

InstanceName	ComponentName
<code>GIC_400</code>	<code>GIC_400</code>
<code>GIC_400.vgic_bus_slave</code>	<code>PVBusSlave</code>

Ports for GIC\_400

Table 3-1023: Ports

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Disable write access to some GIC registers.
cnthpirq[8]	Signal	Slave	Secure physical timer event. PPI interrupt id 26.
cntpnsirq[8]	Signal	Slave	Non-secure physical timer event. PPI interrupt id 30.
cntpsirq[8]	Signal	Slave	Secure physical timer event. PPI interrupt id 29.
cntvirq[8]	Signal	Slave	Virtual timer event. PPI interrupt id 27.
fiqcpu[8]	Signal	Master	FIQ signal to the corresponding processor.
fiqout[8]	Signal	Master	FIQOUT signal to the corresponding processor.
irqcpu[8]	Signal	Master	IRQ signal to the corresponding processor.
irqout[8]	Signal	Master	IRQOUT signal to the corresponding processor.
irqs[480]	Signal	Slave	Interrupt request input lines for the GIC.
legacyfiq[8]	Signal	Slave	Signal into the GIC-400 from the legacy interrupt system. PPI interrupt id 28.
legacyirq[8]	Signal	Slave	Signal into the GIC-400 from the legacy interrupt system. PPI interrupt id 31.
pvbus_s	PVBus	Slave	Handles incoming transactions from PVBus masters.
reset_signal	Signal	Slave	Reset signal input.
vfiqcpu[8]	Signal	Master	Virtual FIQ signal to the processor.
virqcpu[8]	Signal	Master	Virtual IRQ signal to the processor.

Parameters for GIC\_400

NUM\_CPUS

Type

int

Default value

0x1

Description

Number of interfaces to support.

NUM\_SPIS

Type

int

Default value

0xe0

Description

Number of interrupt pins.

**enable\_log\_errors****Type**

bool

**Default value**

0x0

**Description****enable\_log\_fatal****Type**

bool

**Default value**

0x0

**Description****enable\_log\_warnings****Type**

bool

**Default value**

0x0

**Description****interface0.cluster\_id****Type**

int

**Default value**

0x0

**Description**

The CLUSTERID of the interface you want to appear as interface0 in the VGIC.

**interface0.core\_id****Type**

int

**Default value**

0x0

**Description**

The core id of interface0 in the cluster.

**interface0.inout\_port\_number\_to\_use****Type**

int

**Default value**

0x0

**Description**

Which ppiN port is used for this interface.

**interface1.cluster\_id****Type**

int

**Default value**

0x0

**Description**

The CLUSTERID of the interface you want to appear as interface1 in the VGIC.

**interface1.core\_id****Type**

int

**Default value**

0x0

**Description**

The core id of interface1 in the cluster.

**interface1.inout\_port\_number\_to\_use****Type**

int

**Default value**

0x1

**Description**

Which ppiN port is used for this interface.

**interface2.cluster\_id****Type**

int

**Default value**

0x0

**Description**

The CLUSTERID of the interface you want to appear as interface2 in the VGIC.

**interface2.core\_id****Type**

int

**Default value**

0x0

**Description**

The core id of interface2 in the cluster.

**interface2.inout\_port\_number\_to\_use****Type**

int

**Default value**

0x2

**Description**

Which ppiN port is used for this interface.

**interface3.cluster\_id****Type**

int

**Default value**

0x0

**Description**

The CLUSTERID of the interface you want to appear as interface3 in the VGIC.

**interface3.core\_id****Type**

int

**Default value**

0x0

**Description**

The core id of interface3 in the cluster.

**interface3.inout\_port\_number\_to\_use****Type**

int

**Default value**

0x3

**Description**

Which ppiN port is used for this interface.

**interface4.cluster\_id****Type**

int

**Default value**

0x0

**Description**

The CLUSTERID of the interface you want to appear as interface4 in the VGIC.

**interface4.core\_id****Type**

int

**Default value**

0x0

**Description**

The core id of interface4 in the cluster.

**interface4.inout\_port\_number\_to\_use****Type**

int

**Default value**

0x4

**Description**

Which ppiN port is used for this interface.

**interface5.cluster\_id****Type**

int

**Default value**

0x0

**Description**

The CLUSTERID of the interface you want to appear as interface5 in the VGIC.

**interface5.core\_id****Type**

int

**Default value**

0x0

**Description**

The core id of interface5 in the cluster.

**interface5.inout\_port\_number\_to\_use****Type**

int

**Default value**

0x5

**Description**

Which ppiN port is used for this interface.

**interface6.cluster\_id****Type**

int

**Default value**

0x0

**Description**

The CLUSTERID of the interface you want to appear as interface6 in the VGIC.

**interface6.core\_id****Type**

int

**Default value**

0x0

**Description**

The core id of interface6 in the cluster.

**interface6.inout\_port\_number\_to\_use****Type**

int

**Default value**

0x6

**Description**

Which ppiN port is used for this interface.

**interface7.cluster\_id****Type**

int

**Default value**

0x0

**Description**

The CLUSTERID of the interface you want to appear as interface7 in the VGIC.



**interface7.core\_id**

**Type**

int

**Default value**

0x0

**Description**

The core id of interface7 in the cluster.

**interface7.inout\_port\_number\_to\_use**

**Type**

int

**Default value**

0x7

**Description**

Which ppiN port is used for this interface.

3.10.45 GIC\_IRI

GIC metacomponent for redistribution of interrupts. Contains a Distributor and a configurable number of ITSs and Redistributors. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1024: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About GIC\_IRI

The GIC\_IRI has one slave PVBUS interface and one master PVBUS interface. It behaves in a similar manner to a GICv3-compatible device, with the slave interface, `pvbust_s`, granting access to the register banks used by the configuration and operation of MSIs and the master interface, `pvbust_m`, issuing transactions that are required by the ITS and the redistributors in LPI-related operations. All transactions that are routed to the slave port terminate in the component. Accesses to unmapped space are **RAZ/WI**.

An instance of GICv3 requires a small set of parameters to be configured to be useful. For example:

```
gic_iri : GIC_IRI(  
    "reg-base" = 0xF0020000, //Base address for GICD_* REGISTERS, 64K aligned  
    "CPU-affinities" = "0.0.0.0, 0.0.1.0, 0.0.1.1",  
    //A comma-separated list of affinity addresses corresponding to  
    //cpu affinities in the system
```

```
"reg-base-per-redistributor"="0.0.0.0=0xF0040000,0.0.1.0=0xF0060000,
0.0.0.0=0xF0080000",
    //Base addresses for each redistributor in a comma-separated list of
    //affinity=address
);
```

To use LPIs, an ITS must be configured. A minimal configuration might consist of, for example:

```
"ITS-count" = 1, //The number of ITSs in the IRI. Defaults to zero.
"ITS0-base" = 0xF0100000,
"GITS_BASER0-type" = 1, //Type 1 is Devices. A device table is always needed.
"GITS_BASER2-type" = 4, //Type 4 is Collections.
                        //A collection table is needed if GITS_TYPER.HCC is 0.
```

To use GICv4 functionality, one or more ITSs must be configured, as shown in the previous example. In addition, the following parameters are required:

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



Note

- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- Set the `GICD_ITARGETSR-RAZWI` parameter to true for legacy, GICv2-style routing, where interrupts are routed to the first processor in the system.

## Iris and MTI instances for GIC\_IRI

This model has the following Iris instances:

**Table 3-1025: GIC\_IRI Iris instances**

InstanceName	ComponentName
GIC_IRI	GIC_IRI
GIC_IRI.rd_0	GICv3RedistributorInternal
GIC_IRI.rd_0_0	GICv3RedistributorInternal
GIC_IRI.rd_0_0_0	GICv3RedistributorInternal
GIC_IRI.rd_0_0_0_0	GICv3Redistributor
GIC_IRI.rd_t1	GICv3Distributor

This model has the following MTI trace components:

**Table 3-1026: GIC\_IRI MTI instances**

InstanceName	ComponentName
GIC_IRI.rd_0	GICv3RedistributorInternal
GIC_IRI.rd_0_0	GICv3RedistributorInternal
GIC_IRI.rd_0_0_0	GICv3RedistributorInternal

InstanceName	ComponentName
GIC_IRI.rd_0_0_0_0	GICv3Redistributor
GIC_IRI.rd_t1	GICv3Distributor

## Ports for GIC\_IRI

**Table 3-1027: Ports**

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Disable some SPIs signal.
extended_ppi_in_0[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
extended_ppi_in_1[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
extended_ppi_in_10[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
extended_ppi_in_100[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100.
extended_ppi_in_101[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101.
extended_ppi_in_102[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102.
extended_ppi_in_103[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103.
extended_ppi_in_104[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104.
extended_ppi_in_105[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105.
extended_ppi_in_106[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106.
extended_ppi_in_107[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107.
extended_ppi_in_108[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108.
extended_ppi_in_109[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109.
extended_ppi_in_11[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
extended_ppi_in_110[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110.
extended_ppi_in_111[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111.
extended_ppi_in_112[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112.
extended_ppi_in_113[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113.
extended_ppi_in_114[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114.

Name	Protocol	Type	Description
extended_ppi_in_115[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115.
extended_ppi_in_116[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116.
extended_ppi_in_117[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117.
extended_ppi_in_118[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118.
extended_ppi_in_119[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119.
extended_ppi_in_12[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_120[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120.
extended_ppi_in_121[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121.
extended_ppi_in_122[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122.
extended_ppi_in_123[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123.
extended_ppi_in_124[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124.
extended_ppi_in_125[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125.
extended_ppi_in_126[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126.
extended_ppi_in_127[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127.
extended_ppi_in_128[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128.
extended_ppi_in_129[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129.
extended_ppi_in_13[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.
extended_ppi_in_130[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130.
extended_ppi_in_131[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131.
extended_ppi_in_132[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132.
extended_ppi_in_133[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133.
extended_ppi_in_134[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134.
extended_ppi_in_135[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135.

Name	Protocol	Type	Description
extended_ppi_in_136[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136.
extended_ppi_in_137[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137.
extended_ppi_in_138[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138.
extended_ppi_in_139[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139.
extended_ppi_in_14[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.
extended_ppi_in_140[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140.
extended_ppi_in_141[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141.
extended_ppi_in_142[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142.
extended_ppi_in_143[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143.
extended_ppi_in_144[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144.
extended_ppi_in_145[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145.
extended_ppi_in_146[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146.
extended_ppi_in_147[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147.
extended_ppi_in_148[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148.
extended_ppi_in_149[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149.
extended_ppi_in_15[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_150[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150.
extended_ppi_in_151[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151.
extended_ppi_in_152[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152.
extended_ppi_in_153[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153.
extended_ppi_in_154[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154.
extended_ppi_in_155[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155.
extended_ppi_in_156[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156.

Name	Protocol	Type	Description
extended_ppi_in_157[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157.
extended_ppi_in_158[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158.
extended_ppi_in_159[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159.
extended_ppi_in_16[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_160[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160.
extended_ppi_in_161[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161.
extended_ppi_in_162[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162.
extended_ppi_in_163[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163.
extended_ppi_in_164[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164.
extended_ppi_in_165[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165.
extended_ppi_in_166[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166.
extended_ppi_in_167[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167.
extended_ppi_in_168[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168.
extended_ppi_in_169[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169.
extended_ppi_in_17[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_170[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170.
extended_ppi_in_171[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171.
extended_ppi_in_172[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172.
extended_ppi_in_173[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173.
extended_ppi_in_174[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174.
extended_ppi_in_175[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175.
extended_ppi_in_176[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176.
extended_ppi_in_177[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177.

Name	Protocol	Type	Description
extended_ppi_in_178[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178.
extended_ppi_in_179[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179.
extended_ppi_in_18[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_180[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180.
extended_ppi_in_181[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181.
extended_ppi_in_182[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182.
extended_ppi_in_183[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183.
extended_ppi_in_184[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184.
extended_ppi_in_185[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185.
extended_ppi_in_186[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186.
extended_ppi_in_187[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187.
extended_ppi_in_188[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188.
extended_ppi_in_189[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189.
extended_ppi_in_19[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.
extended_ppi_in_190[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190.
extended_ppi_in_191[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191.
extended_ppi_in_192[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192.
extended_ppi_in_193[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193.
extended_ppi_in_194[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194.
extended_ppi_in_195[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195.
extended_ppi_in_196[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196.
extended_ppi_in_197[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197.
extended_ppi_in_198[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198.

Name	Protocol	Type	Description
extended_ppi_in_199[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199.
extended_ppi_in_2[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_20[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_200[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200.
extended_ppi_in_201[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201.
extended_ppi_in_202[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202.
extended_ppi_in_203[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203.
extended_ppi_in_204[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204.
extended_ppi_in_205[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205.
extended_ppi_in_206[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206.
extended_ppi_in_207[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207.
extended_ppi_in_208[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208.
extended_ppi_in_209[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209.
extended_ppi_in_21[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.
extended_ppi_in_210[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210.
extended_ppi_in_211[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211.
extended_ppi_in_212[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212.
extended_ppi_in_213[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213.
extended_ppi_in_214[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214.
extended_ppi_in_215[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215.
extended_ppi_in_216[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216.
extended_ppi_in_217[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217.
extended_ppi_in_218[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218.



Name	Protocol	Type	Description
extended_ppi_in_219[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219.
extended_ppi_in_22[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.
extended_ppi_in_220[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220.
extended_ppi_in_221[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221.
extended_ppi_in_222[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222.
extended_ppi_in_223[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223.
extended_ppi_in_224[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224.
extended_ppi_in_225[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225.
extended_ppi_in_226[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226.
extended_ppi_in_227[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227.
extended_ppi_in_228[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228.
extended_ppi_in_229[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229.
extended_ppi_in_23[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_230[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230.
extended_ppi_in_231[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231.
extended_ppi_in_232[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232.
extended_ppi_in_233[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233.
extended_ppi_in_234[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234.
extended_ppi_in_235[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235.
extended_ppi_in_236[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236.
extended_ppi_in_237[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237.
extended_ppi_in_238[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238.
extended_ppi_in_239[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239.

Name	Protocol	Type	Description
extended_ppi_in_24[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.
extended_ppi_in_240[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240.
extended_ppi_in_241[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241.
extended_ppi_in_242[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242.
extended_ppi_in_243[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243.
extended_ppi_in_244[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244.
extended_ppi_in_245[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245.
extended_ppi_in_246[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246.
extended_ppi_in_247[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247.
extended_ppi_in_248[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248.
extended_ppi_in_249[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249.
extended_ppi_in_25[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_250[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250.
extended_ppi_in_251[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251.
extended_ppi_in_252[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252.
extended_ppi_in_253[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253.
extended_ppi_in_254[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254.
extended_ppi_in_255[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255.
extended_ppi_in_26[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.
extended_ppi_in_28[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_3[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.

Name	Protocol	Type	Description
extended_ppi_in_30[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.
extended_ppi_in_31[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.
extended_ppi_in_32[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.
extended_ppi_in_35[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.
extended_ppi_in_36[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_4[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_40[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.
extended_ppi_in_42[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.
extended_ppi_in_43[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.
extended_ppi_in_47[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.
extended_ppi_in_49[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_5[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_50[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.

Name	Protocol	Type	Description
extended_ppi_in_51[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.
extended_ppi_in_52[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.
extended_ppi_in_54[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.
extended_ppi_in_56[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_6[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_60[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.
extended_ppi_in_64[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64.
extended_ppi_in_65[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65.
extended_ppi_in_66[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66.
extended_ppi_in_67[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67.
extended_ppi_in_68[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68.
extended_ppi_in_69[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69.
extended_ppi_in_7[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_70[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70.
extended_ppi_in_71[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71.

Name	Protocol	Type	Description
extended_ppi_in_72[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72.
extended_ppi_in_73[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73.
extended_ppi_in_74[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74.
extended_ppi_in_75[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75.
extended_ppi_in_76[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76.
extended_ppi_in_77[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77.
extended_ppi_in_78[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78.
extended_ppi_in_79[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79.
extended_ppi_in_8[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.
extended_ppi_in_80[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80.
extended_ppi_in_81[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81.
extended_ppi_in_82[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82.
extended_ppi_in_83[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83.
extended_ppi_in_84[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84.
extended_ppi_in_85[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85.
extended_ppi_in_86[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86.
extended_ppi_in_87[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87.
extended_ppi_in_88[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88.
extended_ppi_in_89[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89.
extended_ppi_in_9[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.
extended_ppi_in_90[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90.
extended_ppi_in_91[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91.
extended_ppi_in_92[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92.

Name	Protocol	Type	Description
extended_ppi_in_93[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93.
extended_ppi_in_94[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94.
extended_ppi_in_95[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95.
extended_ppi_in_96[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96.
extended_ppi_in_97[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97.
extended_ppi_in_98[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98.
extended_ppi_in_99[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99.
extended_spi_in[1024]	Signal	Slave	Extended Shared peripheral interrupts.
msi_error_interrupt	Signal	Master	When report of MSI error allowed through interrupt, loop-back this signal to relevant IRQ input signal
po_reset	Signal	Slave	Resets.
ppi_in_0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_10[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_100[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 106.
ppi_in_107[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 107.
ppi_in_108[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 109.
ppi_in_11[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_110[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 113.
ppi_in_114[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 114.
ppi_in_115[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 118.

Name	Protocol	Type	Description
ppi_in_119[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_12[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_120[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 120.
ppi_in_121[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 122.
ppi_in_123[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 123.
ppi_in_124[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 126.
ppi_in_127[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 128.
ppi_in_129[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_13[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_130[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_14[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_140[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 141.
ppi_in_142[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 142.
ppi_in_143[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 145.
ppi_in_146[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 149.
ppi_in_15[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_150[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 150.
ppi_in_151[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 153.

Name	Protocol	Type	Description
ppi_in_154[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 157.
ppi_in_158[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 159.
ppi_in_16[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_160[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 162.
ppi_in_163[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 164.
ppi_in_165[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_17[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_170[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 177.
ppi_in_178[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 178.
ppi_in_179[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_18[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_180[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 181.
ppi_in_182[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 184.
ppi_in_185[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 187.
ppi_in_188[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 189.



Name	Protocol	Type	Description
ppi_in_19[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_190[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 193.
ppi_in_194[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 194.
ppi_in_195[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 195.
ppi_in_196[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 199.
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_20[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_200[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_21[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_210[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 211.
ppi_in_212[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 212.
ppi_in_213[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 217.
ppi_in_218[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 219.
ppi_in_22[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_220[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 222.
ppi_in_223[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 223.

Name	Protocol	Type	Description
ppi_in_224[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 229.
ppi_in_23[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_230[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 230.
ppi_in_231[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 234.
ppi_in_235[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 236.
ppi_in_237[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_24[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_240[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 247.
ppi_in_248[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 248.
ppi_in_249[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_25[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_250[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 253.
ppi_in_254[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 254.
ppi_in_255[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_26[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 29.

Name	Protocol	Type	Description
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_30[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_40[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_50[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_60[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 64.

Name	Protocol	Type	Description
ppi_in_65[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_70[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_8[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_80[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_9[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 9.
ppi_in_90[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 95.
ppi_in_96[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 99.
pvbus_m	PVBus	Master	Memory bus out: transactions generated by the IRI.

Name	Protocol	Type	Description
pvbus_s	PVBus	Slave	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Resets.
spi_in[988]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.
wire_to_msi_in_0[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 0.
wire_to_msi_in_1[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 1.
wire_to_msi_in_2[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 2.
wire_to_msi_in_3[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 3.

## Parameters for GIC\_IRI

### A3-affinity-supported

#### Type

bool

#### Default value

0x0

#### Description

Device supports affinity level 3 values that are non-zero.

### ARE-fixed-to-one

#### Type

bool

#### Default value

0x0

#### Description

GICv2 compatibility is not supported and GICD\_CTLR.ARE\_\* is always one.

### CPU-affinities

#### Type

string

#### Default value

0.0.0.0

#### Description

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

**CPU-affinities-file****Type**

string

**Default value**

""

**Description**

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

**DPG-ARE-only****Type**

bool

**Default value**

0x0

**Description**

Limit application of DPG bits to interrupt groups for which ARE=1.

**DPG-bits-implemented****Type**

bool

**Default value**

0x0

**Description**

Enable implementation of interrupt group participation bits or DPG bits in GICR\_CTLR.

**DS-fixed-to-zero****Type**

bool

**Default value**

0x0

**Description**

Enable/disable support of single security state.

**GICD-alias****Type**

int

**Default value**

0x0

**Description**

In GICv2 mode: the base address for a 4k page alias of the first 4k of the Distributor page, in GICv3 mode. the base address of a 64KB page containing message based SPI signalling register aliases(0:Disabled).

**GICD-legacy-registers-as-reserved****Type**

bool

**Default value**

0x0

**Description**

When ARE is RAO/WI, makes superfluous registers in GICD reserved ( including for the purpose of STATUSR updates).

**GICD\_CTLR-DS-1-means-secure-only****Type**

bool

**Default value**

0x0

**Description**

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

**GICD\_ITARGETSR-RAZWI****Type**

bool

**Default value**

0x0

**Description**

If true, the GICD\_ITARGETS registers are RAZ/WI.

**GICD\_PIDR****Type**

int

**Default value**

0x0

**Description**

The value for the GICD\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

**GICD\_TYPER2****Type**

int

**Default value**

0x0

**Description**

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

**GICR-clear-enable-supported****Type**

bool

**Default value**

0x0

**Description**

When true, this sets the value of the RO bit GICR\_CTLR.CES with the value of the parameter allow-LPIEN-clear, making it visible to software.

**GICR-invalidate-registers-implemented****Type**

bool

**Default value**

0x0

**Description**

When true, the registers GICR\_INVLPIR, GICR\_INVALLR and GICR\_SYNCR are implemented.

**GICR\_PIDR****Type**

int

**Default value**

0x0

**Description**

The value for the GICR\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

**GICR\_PROPBASER-read-only****Type**

bool

**Default value**

0x0



**Description**

GICR\_PROPBASER register is read-only.

**GICR\_PROPBASER-reset-value****Type**

int

**Default value**

0x0

**Description**

Value of GICR\_PROPBASER on reset.

**GITS\_BASER0-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER0 register.

**GITS\_BASER0-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER0 register is RAZ/WI.

**GITS\_BASER0-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER0 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER1-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER1 register.

**GITS\_BASER1-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER1 register is RAZ/WI.

**GITS\_BASER1-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER2-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER2 register.

**GITS\_BASER2-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER2 register is RAZ/WI.

**GITS\_BASER2-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER3-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER3 register.

**GITS\_BASER3-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER3 register is RAZ/WI.

**GITS\_BASER3-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER3 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER4-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER4 register.

**GITS\_BASER4-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER4 register is RAZ/WI.

**GITS\_BASER4-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER4 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER5-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER5 register.

**GITS\_BASER5-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER5 register is RAZ/WI.

**GITS\_BASER5-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER5 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER6-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER6 register.

**GITS\_BASER6-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER6 register is RAZ/WI.

**GITS\_BASER6-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER6 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER7-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER7 register.

**GITS\_BASER7-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER7 register is RAZ/WI.

**GITS\_BASER7-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER7 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_PIDR****Type**

int

**Default value**

0x0

**Description**

The value for the GITS\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

**ICFGR-PPI-mask****Type**

int

**Default value**

0xaaaaaaaa

**Description**

Mask for writes to ICFGR registers that configure PPIs.

**ICFGR-PPI-reset****Type**

int

**Default value**

0x0

**Description**

Reset value for ICFGR registers that configure PPIs.

**ICFGR-SGI-mask****Type**

int

**Default value**

0x0

**Description**

Mask for writes to ICFGR registers that configure SGIs.

**ICFGR-SGI-reset****Type**

int

**Default value**

0xaaaaaaaa

**Description**

Reset value for ICFGR registers that configure SGIs.

**ICFGR-SPI-mask****Type**

int

**Default value**

0xaaaaaaaa

**Description**

Mask for writes to ICFGR registers that configure SPIs.

**ICFGR-SPI-reset****Type**

int

**Default value**

0x0

**Description**

Reset value for ICFGR registers that configure SPIs.

**ICFGR-rsvd-bit****Type**

bool

**Default value**

0x0

**Description**

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGRn for n>0.

**IGROUP-PPI-mask****Type**

int

**Default value**

0xffff

**Description**

Mask for writes to PPI bits in IGROUP registers.

**IGROUP-PPI-reset****Type**

int

**Default value**

0x0

**Description**

Reset value for SGI bits in IGROUP registers.

**IGROUP-SGI-mask****Type**

int

**Default value**

0xffff

**Description**

Mask for writes to SGI bits in IGROUP registers.

**IGROUP-SGI-reset****Type**

int

**Default value**

0x0

**Description**

Reset value for SGI bits in IGROUP registers.

**IIDR****Type**

int

**Default value**

0x0



**Description**

GICD\_IIDR and GICR\_IIDR value.

**IRI-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits used to represent interrupts IDs in the Distributor and Redistributors, forced to 10 when none of LPIs, extended SPIs or extended PPIs is supported.

**IROUTER-IRM-RAZ-WI****Type**

bool

**Default value**

0x0

**Description**

GICD\_IROUTERn.InterruptRoutingMode is RAZ/WI.

**ITS-BASER-force-page-alignment****Type**

bool

**Default value**

0x1

**Description**

Force alignment of address written to a GITS\_BASER register to the page size configured.

**ITS-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of interrupt bits supported by ITS.

**ITS-MOVALL-update-collections****Type**

bool

**Default value**

0x0

**Description**

Whether MOVALL command updates the collection entires.

**ITS-TRANSLATE64R****Type**

bool

**Default value**

0x0

**Description**

Add an implementation specific register at 0x10008 supporting 64 bit TRANSLATER (dev[63:32], interrupt[31:0]).

**ITS-cache-invalidate-on-disable****Type**

bool

**Default value**

0x0

**Description**

Sets the RO bit GITS\_TYPER.INV. When true, after the following sequence: 1) GITS\_CTLR.Enabled written 1-->0, 2) GITS\_CTLR.Quiescent observed as 1, 3) GITS\_BASER<n>.Valid written 1-->0, there is no cached information from the ITS memory structure pointed to by GITS\_BASER<n>.

**ITS-collection-ID-bits****Type**

int

**Default value**

0x0

**Description**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0.

**ITS-count****Type**

int

**Default value**

0x0

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**ITS-cumulative-collection-tables****Type**

bool

**Default value**

0x1

**Description**

When true, the supported amount of collections is the sum of GITS\_TYPER.HCC and the number of collections supported in memory, otherwise, simply the number supported in memory only. Irrelevant when HCC=0.

**ITS-device-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits supported for ITS device IDs.

**ITS-enable-itt-address-verification****Type**

bool

**Default value**

0x0

**Description**

If true, a transaction will be sent to ITT Address for verification.

**ITS-entry-size****Type**

int

**Default value**

0x8

**Description**

Number of bytes required to store each entry in the ITT tables.

**ITS-hardware-collection-count****Type**

int

**Default value**

0x0

**Description**

Number of hardware collections held exclusively in the ITS.

**ITS-legacy-iidr-typer-offset****Type**

bool

**Default value**

0x0

**Description**

Put the GITS\_IIDR and GITS\_TYPER registers at their older offset of 0x8 and 0x4 respectively.

**ITS-shared-vPE-table****Type**

int

**Default value**

0x0

**Description**

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

**ITS-threaded-command-queue****Type**

bool

**Default value**

0x1

**Description**

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

**ITS-use-physical-target-addresses****Type**

bool

**Default value**

0x1

**Description**

Use physical hardware addresses for targets in ITS commands -- must be true for distributed implementations.

**ITS-vmovp-bit****Type**

bool

**Default value**

0x0

**Description**

Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

**ITS0-base****Type**

int

**Default value**

0x0

**Description**

Register base address for ITS0 (automatic if 0).

**ITS1-base****Type**

int

**Default value**

0x0

**Description**

Register base address for ITS1 (automatic if 0).

**ITS2-base****Type**

int

**Default value**

0x0

**Description**

Register base address for ITS2 (automatic if 0).

**ITS3-base****Type**

int

**Default value**

0x0

**Description**

Register base address for ITS3 (automatic if 0).

**LPI-cache-check-data****Type**

bool

**Default value**

0x0

**Description**

Enable Cached LPI data against memory checking when available for cache type.

**LPI-cache-type****Type**

int

**Default value**

0x1

**Description**

Cache type for LPIs, 0:No caching, 1:Full caching.

**MSI\_IIDR****Type**

int

**Default value**

0x0

**Description**

Value returned in MSI\_IIDR registers.

**MSI\_NS-frame0-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 0 registers.

**MSI\_NS-frame0-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

**MSI\_NS-frame0-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

**MSI\_NS-frame1-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 1 registers.

**MSI\_NS-frame1-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

**MSI\_NS-frame1-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

**MSI\_NS-frame2-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 2 registers.

**MSI\_NS-frame2-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

**MSI\_NS-frame2-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

**MSI\_NS-frame3-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 3 registers.

**MSI\_NS-frame3-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

**MSI\_NS-frame3-min-SPI****Type**

int

**Default value**

0x0



**Description**

Minimum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

**MSI\_NS-frame4-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 4 registers.

**MSI\_NS-frame4-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

**MSI\_NS-frame4-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

**MSI\_NS-frame5-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 5 registers.

**MSI\_NS-frame5-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

**MSI\_NS-frame5-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

**MSI\_NS-frame6-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 6 registers.

**MSI\_NS-frame6-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

**MSI\_NS-frame6-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

**MSI\_NS-frame7-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 7 registers.

**MSI\_NS-frame7-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

**MSI\_NS-frame7-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

**MSI\_PIDR****Type**

int

**Default value**

0x0

**Description**

The value for the MSI\_PIDR registers, if non-zero and distributor supports GICv2m. Note: fixed fields (device type etc.) will be overridden in this value.

**MSI\_S-frame0-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 0 registers.

**MSI\_S-frame0-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

**MSI\_S-frame0-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

**MSI\_S-frame1-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 1 registers.

**MSI\_S-frame1-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

**MSI\_S-frame1-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

**MSI\_S-frame2-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 2 registers.

**MSI\_S-frame2-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

**MSI\_S-frame2-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

**MSI\_S-frame3-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 3 registers.

**MSI\_S-frame3-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

**MSI\_S-frame3-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

**MSI\_S-frame4-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 4 registers.

**MSI\_S-frame4-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

**MSI\_S-frame4-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

**MSI\_S-frame5-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 5 registers.

**MSI\_S-frame5-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

**MSI\_S-frame5-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

**MSI\_S-frame6-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 6 registers.

**MSI\_S-frame6-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

**MSI\_S-frame6-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

**MSI\_S-frame7-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 7 registers.

**MSI\_S-frame7-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

**MSI\_S-frame7-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

**PA\_SIZE****Type**

int

**Default value**

0x30

**Description**

Number of valid bits in physical address.

**PPI-implemented-mask****Type**

int

**Default value**

0xffff

**Description**

Mask of PPIs that are implemented. One bit per PPI bit 0 == PPI 16 (first PPI). This will affect other masks.



**SPI-count****Type**

int

**Default value**

0xe0

**Description**

Number of SPIs that are implemented.

**SPI-message-based-support****Type**

bool

**Default value**

0x1

**Description**

Distributor supports message based signaling of SPI.

**SPI-unimplemented****Type**

string

**Default value**

""

**Description**

A comma separated list of unimplemented SPIs ranges for sparse SPI definition(for ex: '35, 39-42, 73').

**STATUSR-implemented****Type**

bool

**Default value**

0x1

**Description**

Determines whether the GICR\_STATUSR register is implemented.

**add-output-cpu-wake-request-signal-from-redistributor****Type**

bool

**Default value**

0x0

**Description**

if true, the redistributor will have the output signal `cpu_wake_request` from GIC to DSU and if false, the signals are not added to the redistributor.

**allow-LPIEN-clear****Type**

bool

**Default value**

0x0

**Description**

Allow RW behaviour on GICR\_CTLR.LPIEN instead of set once.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

**common-lpi-configuration****Type**

int

**Default value**

0x0

**Description**

Describes which re-distributors share (and must be configured with the same) LPI configuration table as described in GICR\_TYPER( 0:All, 1:A.x.x.x, 2:A.B.x.x, 3:A.B.C.x.

**common-vPE-table-affinity****Type**

string

**Default value**

""

**Description**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s) (where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on). Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when `has-gicv4.1` is true.

**consolidators****Type**

string

**Default value**

""

**Description**

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form 'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1, [etc]'. (eg '0x3f100000:64:4096, 0x3f200000:64:4224'). The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

**delay-ITS-accesses****Type**

bool

**Default value**

0x1

**Description**

Delay accesses from the ITS until GICR\_SYNCRR is read.

**delay-redistributor-accesses****Type**

bool

**Default value**

0x1

**Description**

Delay memory accesses from the redistributor until GICR\_SYNCRR is read.

**direct-lpi-support****Type**

bool

**Default value**

0x0

**Description**

Enable support for LPI operations through GICR registers.

**enable\_protocol\_checking****Type**

bool

**Default value**

0x0

**Description**

Enable/disable protocol checking at cpu interface.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**extended-ppi-count****Type**

int

**Default value**

0x0

**Description**

Number of extended PPI supported.

**extended-spi-count****Type**

int

**Default value**

0x0

**Description**

Number of extended SPI supported.

**fixed-routed-spis****Type**

string

**Default value**

""

**Description**

Value of IROUTER[n] register in the form 'n=a.b.c.d, n=\*'. The RM bit of IROUTER is 0 when n=a.b.c.d is used else 1 when n=\* is used. n can be >= 32 and <= 1019.

**gicr-icfgr-extended-count****Type**

int

**Default value**

0x4

**Description**

Number of extended GICR\_ICFGR registers supported.

**`gicv2-only`****Type**

bool

**Default value**

0x0

**Description**

If true, when using the GICv3 model, pretend to be a GICv2 system.

**`group-enables-control-doorbell`****Type**

bool

**Default value**

0x0

**Description**

When true, GICR\_VPENDBASER.{VGrp0En,VGrp1En} are cached to allow GIC to check group enables when virtual interrupt targeting this VCPU which is non-resident reaches Redistributor.

**`has-gicv4.1`****Type**

bool

**Default value**

0x0

**Description**

Enable GICv4.1 functionality; when false the component is inactive.

**`has-two-security-states`****Type**

bool

**Default value**

0x1

**Description**

If true, has two security states.

**has\_VPENDBASER-dirty-flag-on-load****Type**

bool

**Default value**

0x0

**Description**

GICR\_VPENDBASER.Dirty reflects transient loading state when valid=1.

**has\_mpam****Type**

bool

**Default value**

0x0

**Description**

Enable MPAM support on ITS and RDs.

**has\_nmi****Type**

bool

**Default value**

0x0

**Description**

Enable support for Non-maskable Interrupts (NMIs). (FEAT\_GICv3\_NMI).

**ignore-generate-sgi-when-no-are****Type**

bool

**Default value**

0x0

**Description**

Ignore GenerateSgi packets coming from the CPU interface if both ARE\_S and ARE\_NS are 0.

**individual-doorbell-not-supported****Type**

bool

**Default value**

0x0

**Description**

For IRI with support of virtual interrupt, individual doorbell is not supported when true.

**irouter-default-mask****Type**

string

**Default value**

""

**Description**

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

**irouter-default-reset****Type**

string

**Default value**

""

**Description**

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or '\*'.

**irouter-mask-values****Type**

string

**Default value**

""

**Description**

Mask Value of IROUTER[n] register in the form 'n=a.b.c.d'.n can be >= 32 and <= 1019.

**irouter-reset-values****Type**

string

**Default value**

""

**Description**

Reset Value of IROUTER[n] register in the form 'n=a.b.c.d or n=\*'.n can be >= 32 and <= 1019.

**legacy-sgi-enable-rao****Type**

bool

**Default value**

0x0

**Description**

Enables for SGI associated with an ARE=0 regime are RAO/WI.

**local-SEIs****Type**

bool

**Default value**

0x0

**Description**

Generate SEI to signal internal issues.

**local-VSEIs****Type**

bool

**Default value**

0x0

**Description**

Generate VSEI to signal internal issues.

**lockable-SPI-count****Type**

int

**Default value**

0x0

**Description**

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

**monolithic****Type**

bool

**Default value**

0x0

**Description**

Indicate that the implementation is not distributed.



**mpam\_max\_partid****Type**

int

**Default value**

0xffff

**Description**

Maximum valid PARTID.

**mpam\_max\_pmg****Type**

int

**Default value**

0xff

**Description**

Maximum valid PMG.

**non-ARE-core-count****Type**

int

**Default value**

0x8

**Description**

Maximum number of non-ARE cores; normally used to pass the cluster-level NUM\_CORES parameter to the top-level redistributor.

**outer-cacheability-support****Type**

bool

**Default value**

0x0

**Description**

Allow configuration of outer cachability attributes in ITS and Redistributor.

**output\_attributes****Type**

string

**Default value**"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID,  
ExtendedID[38]=MPAM\_NS"

**Description**

User-defined transform to be applied to bus attributes like MasterID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**priority-bits****Type**

int

**Default value**

0x5

**Description**

Number of implemented priority bits.

**processor-numbers****Type**

string

**Default value**

""

**Description**

Specify processor numbers (as appears in GICR\_TYPER) in the form 0.0.0.0=0,0.0.0.1=1 etc.)  
If not specified, will number processors starting at 0.

**redistributor-threaded-sync****Type**

bool

**Default value**

0x1

**Description**

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

**reg-base****Type**

int

**Default value**

0x2c010000

**Description**

Base for decoding GICv3 registers.

**reg-base-per-redistributor****Type**

string

**Default value**{}  
  
Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the reg-base parameter (except that reg-base will still be used for the top-level redistributor). If reg-base-per-redistributor-file is specified, this parameter is ignored.

**reg-base-per-redistributor-file****Type**

string

**Default value**

""

**Description**

Path to file containing the base address for each redistributor in the form '0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000'. All redistributors must be specified and this overrides the reg-base parameter (except that reg-base will still be used for the top-level redistributor). If this parameter is specified, reg-base-per-redistributor parameter will be ignored even when it is given.

**report-MSI-error-via-statusr****Type**

int

**Default value**

0x0

**Description**

Report MSI error via GITS\_STATUSR. (0:unsupported, 1:report by GITS\_STATUSR, 2:report by GITS\_STATUSR and interrupt as well).

**rme\_default\_mecid\_nonsecure****Type**

int

**Default value**

0x0

**Description**

Default MECID value for NON-SECURE PAS.

**sgi-range-selector-support****Type**

bool

**Default value**

0x0

**Description**

Device has support for the Range Selector feature for SGI.

**single-set-support****Type**

bool

**Default value**

0x0

**Description**

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

**supports-shareability****Type**

bool

**Default value**

0x1

**Description**

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

**trace-speculative-lpi-property-update****Type**

bool

**Default value**

0x0

**Description**

Trace LPI property updates performed on speculative accesses (useful for debugging LPI).

**vPE-table-entry-size-in-doubleword****Type**

int

**Default value**

0x5

**Description**

The size of one entry of a vPE configuration table in double word. The value decremented by one is shown at GICR\_VPROPBASER.Entry\_Size. Current model mandates the minimum entry size to be 4 doublewords. When lower value is given, it is truncated to 4.

**virtual-lpi-support****Type**

bool

**Default value**

0x0

**Description**

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

**virtual-priority-bits****Type**

int

**Default value**

0x5

**Description**

Number of implemented virtual priority bits.

**wakeup-on-reset****Type**

bool

**Default value**

0x0

## Description

Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

### 3.10.46 GIC\_IRI\_Filter

GIC metacomponent for redistribution of interrupts. Contains a Distributor and a configurable number of ITSs and Redistributors. Validation only version. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1028: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About GIC\_IRI\_Filter

The GIC\_IRI\_Filter has similar behavior to the GIC\_IRI, except for the slave interface. Any transaction accessing a 4 KB page that is not used by the GIC, as configurable through the parameters, is forwarded to the `pvbush_filtermiss_m` port, which is only present in this variant.



**Note**

- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- Set the `GICD_ITARGETSR-RAZWI` parameter to true for legacy, GICv2-style routing, where interrupts are routed to the first processor in the system.

## Iris and MTI instances for GIC\_IRI\_Filter

This model has the following Iris instances:

**Table 3-1029: GIC\_IRI\_Filter Iris instances**

InstanceName	ComponentName
GIC_IRI_Filter	GIC_IRI
GIC_IRI_Filter.rd_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0_0_0	GICv3Redistributor
GIC_IRI_Filter.rd_t1	GICv3Distributor

This model has the following MTI trace components:

**Table 3-1030: GIC\_IRI\_Filter MTI instances**

InstanceName	ComponentName
GIC_IRI_Filter.rd_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0_0_0	GICv3Redistributor
GIC_IRI_Filter.rd_tl	GICv3Distributor

### Ports for GIC\_IRI\_Filter

**Table 3-1031: Ports**

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Disable some SPLs signal.
extended_ppi_in_0[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
extended_ppi_in_1[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
extended_ppi_in_10[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
extended_ppi_in_100[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100.
extended_ppi_in_101[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101.
extended_ppi_in_102[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102.
extended_ppi_in_103[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103.
extended_ppi_in_104[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104.
extended_ppi_in_105[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105.
extended_ppi_in_106[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106.
extended_ppi_in_107[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107.
extended_ppi_in_108[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108.
extended_ppi_in_109[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109.
extended_ppi_in_11[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
extended_ppi_in_110[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110.
extended_ppi_in_111[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111.

Name	Protocol	Type	Description
extended_ppi_in_112[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112.
extended_ppi_in_113[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113.
extended_ppi_in_114[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114.
extended_ppi_in_115[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115.
extended_ppi_in_116[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116.
extended_ppi_in_117[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117.
extended_ppi_in_118[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118.
extended_ppi_in_119[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119.
extended_ppi_in_12[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_120[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120.
extended_ppi_in_121[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121.
extended_ppi_in_122[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122.
extended_ppi_in_123[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123.
extended_ppi_in_124[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124.
extended_ppi_in_125[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125.
extended_ppi_in_126[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126.
extended_ppi_in_127[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127.
extended_ppi_in_128[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128.
extended_ppi_in_129[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129.
extended_ppi_in_13[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.
extended_ppi_in_130[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130.
extended_ppi_in_131[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131.
extended_ppi_in_132[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132.



Name	Protocol	Type	Description
extended_ppi_in_133[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133.
extended_ppi_in_134[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134.
extended_ppi_in_135[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135.
extended_ppi_in_136[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136.
extended_ppi_in_137[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137.
extended_ppi_in_138[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138.
extended_ppi_in_139[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139.
extended_ppi_in_14[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.
extended_ppi_in_140[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140.
extended_ppi_in_141[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141.
extended_ppi_in_142[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142.
extended_ppi_in_143[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143.
extended_ppi_in_144[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144.
extended_ppi_in_145[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145.
extended_ppi_in_146[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146.
extended_ppi_in_147[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147.
extended_ppi_in_148[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148.
extended_ppi_in_149[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149.
extended_ppi_in_15[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_150[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150.
extended_ppi_in_151[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151.
extended_ppi_in_152[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152.
extended_ppi_in_153[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153.

Name	Protocol	Type	Description
extended_ppi_in_154[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154.
extended_ppi_in_155[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155.
extended_ppi_in_156[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156.
extended_ppi_in_157[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157.
extended_ppi_in_158[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158.
extended_ppi_in_159[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159.
extended_ppi_in_16[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_160[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160.
extended_ppi_in_161[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161.
extended_ppi_in_162[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162.
extended_ppi_in_163[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163.
extended_ppi_in_164[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164.
extended_ppi_in_165[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165.
extended_ppi_in_166[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166.
extended_ppi_in_167[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167.
extended_ppi_in_168[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168.
extended_ppi_in_169[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169.
extended_ppi_in_17[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_170[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170.
extended_ppi_in_171[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171.
extended_ppi_in_172[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172.
extended_ppi_in_173[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173.
extended_ppi_in_174[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174.

Name	Protocol	Type	Description
extended_ppi_in_175[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175.
extended_ppi_in_176[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176.
extended_ppi_in_177[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177.
extended_ppi_in_178[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178.
extended_ppi_in_179[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179.
extended_ppi_in_18[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_180[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180.
extended_ppi_in_181[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181.
extended_ppi_in_182[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182.
extended_ppi_in_183[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183.
extended_ppi_in_184[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184.
extended_ppi_in_185[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185.
extended_ppi_in_186[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186.
extended_ppi_in_187[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187.
extended_ppi_in_188[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188.
extended_ppi_in_189[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189.
extended_ppi_in_19[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.
extended_ppi_in_190[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190.
extended_ppi_in_191[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191.
extended_ppi_in_192[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192.
extended_ppi_in_193[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193.
extended_ppi_in_194[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194.
extended_ppi_in_195[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195.

Name	Protocol	Type	Description
extended_ppi_in_196[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196.
extended_ppi_in_197[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197.
extended_ppi_in_198[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198.
extended_ppi_in_199[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199.
extended_ppi_in_2[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_20[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_200[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200.
extended_ppi_in_201[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201.
extended_ppi_in_202[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202.
extended_ppi_in_203[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203.
extended_ppi_in_204[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204.
extended_ppi_in_205[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205.
extended_ppi_in_206[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206.
extended_ppi_in_207[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207.
extended_ppi_in_208[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208.
extended_ppi_in_209[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209.
extended_ppi_in_21[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.
extended_ppi_in_210[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210.
extended_ppi_in_211[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211.
extended_ppi_in_212[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212.
extended_ppi_in_213[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213.
extended_ppi_in_214[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214.
extended_ppi_in_215[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215.

Name	Protocol	Type	Description
extended_ppi_in_216[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216.
extended_ppi_in_217[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217.
extended_ppi_in_218[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218.
extended_ppi_in_219[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219.
extended_ppi_in_22[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.
extended_ppi_in_220[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220.
extended_ppi_in_221[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221.
extended_ppi_in_222[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222.
extended_ppi_in_223[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223.
extended_ppi_in_224[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224.
extended_ppi_in_225[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225.
extended_ppi_in_226[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226.
extended_ppi_in_227[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227.
extended_ppi_in_228[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228.
extended_ppi_in_229[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229.
extended_ppi_in_23[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_230[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230.
extended_ppi_in_231[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231.
extended_ppi_in_232[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232.
extended_ppi_in_233[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233.
extended_ppi_in_234[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234.
extended_ppi_in_235[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235.
extended_ppi_in_236[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236.

Name	Protocol	Type	Description
extended_ppi_in_237[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237.
extended_ppi_in_238[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238.
extended_ppi_in_239[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239.
extended_ppi_in_24[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.
extended_ppi_in_240[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240.
extended_ppi_in_241[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241.
extended_ppi_in_242[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242.
extended_ppi_in_243[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243.
extended_ppi_in_244[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244.
extended_ppi_in_245[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245.
extended_ppi_in_246[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246.
extended_ppi_in_247[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247.
extended_ppi_in_248[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248.
extended_ppi_in_249[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249.
extended_ppi_in_25[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_250[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250.
extended_ppi_in_251[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251.
extended_ppi_in_252[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252.
extended_ppi_in_253[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253.
extended_ppi_in_254[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254.
extended_ppi_in_255[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255.
extended_ppi_in_26[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.

Name	Protocol	Type	Description
extended_ppi_in_28[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_3[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.
extended_ppi_in_30[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.
extended_ppi_in_31[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.
extended_ppi_in_32[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.
extended_ppi_in_35[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.
extended_ppi_in_36[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_4[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_40[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.
extended_ppi_in_42[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.
extended_ppi_in_43[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.
extended_ppi_in_47[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.

Name	Protocol	Type	Description
extended_ppi_in_49[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_5[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_50[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.
extended_ppi_in_51[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.
extended_ppi_in_52[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.
extended_ppi_in_54[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.
extended_ppi_in_56[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_6[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_60[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.
extended_ppi_in_64[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64.
extended_ppi_in_65[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65.
extended_ppi_in_66[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66.
extended_ppi_in_67[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67.
extended_ppi_in_68[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68.
extended_ppi_in_69[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69.



Name	Protocol	Type	Description
extended_ppi_in_7[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_70[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70.
extended_ppi_in_71[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71.
extended_ppi_in_72[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72.
extended_ppi_in_73[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73.
extended_ppi_in_74[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74.
extended_ppi_in_75[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75.
extended_ppi_in_76[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76.
extended_ppi_in_77[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77.
extended_ppi_in_78[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78.
extended_ppi_in_79[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79.
extended_ppi_in_8[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.
extended_ppi_in_80[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80.
extended_ppi_in_81[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81.
extended_ppi_in_82[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82.
extended_ppi_in_83[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83.
extended_ppi_in_84[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84.
extended_ppi_in_85[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85.
extended_ppi_in_86[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86.
extended_ppi_in_87[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87.
extended_ppi_in_88[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88.
extended_ppi_in_89[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89.
extended_ppi_in_9[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.

Name	Protocol	Type	Description
extended_ppi_in_90[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90.
extended_ppi_in_91[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91.
extended_ppi_in_92[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92.
extended_ppi_in_93[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93.
extended_ppi_in_94[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94.
extended_ppi_in_95[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95.
extended_ppi_in_96[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96.
extended_ppi_in_97[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97.
extended_ppi_in_98[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98.
extended_ppi_in_99[64]	Signal	Slave	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99.
extended_spi_in[1024]	Signal	Slave	Extended Shared peripheral interrupts.
msi_error_interrupt	Signal	Master	When report of MSI error allowed through interrupt, loop-back this signal to relevant IRQ input signal
po_reset	Signal	Slave	Resets.
ppi_in_0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_10[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_100[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 106.
ppi_in_107[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 107.
ppi_in_108[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 109.
ppi_in_11[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_110[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 113.

Name	Protocol	Type	Description
ppi_in_114[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 114.
ppi_in_115[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 118.
ppi_in_119[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_12[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_120[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 120.
ppi_in_121[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 122.
ppi_in_123[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 123.
ppi_in_124[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 126.
ppi_in_127[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 128.
ppi_in_129[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_13[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_130[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_14[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_140[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 141.
ppi_in_142[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 142.
ppi_in_143[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 145.
ppi_in_146[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 149.

Name	Protocol	Type	Description
ppi_in_15[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_150[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 150.
ppi_in_151[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 153.
ppi_in_154[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 157.
ppi_in_158[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 159.
ppi_in_16[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_160[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 162.
ppi_in_163[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 164.
ppi_in_165[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_17[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_170[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 177.
ppi_in_178[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 178.
ppi_in_179[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_18[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_180[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 181.
ppi_in_182[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 184.

Name	Protocol	Type	Description
ppi_in_185[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 187.
ppi_in_188[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 189.
ppi_in_19[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_190[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 193.
ppi_in_194[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 194.
ppi_in_195[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 195.
ppi_in_196[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 199.
ppi_in_2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_20[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_200[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_21[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_210[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 211.
ppi_in_212[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 212.
ppi_in_213[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 217.
ppi_in_218[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 219.

Name	Protocol	Type	Description
ppi_in_22[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_220[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 222.
ppi_in_223[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 223.
ppi_in_224[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 229.
ppi_in_23[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_230[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 230.
ppi_in_231[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 234.
ppi_in_235[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 236.
ppi_in_237[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_24[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_240[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 247.
ppi_in_248[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 248.
ppi_in_249[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_25[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_250[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 253.
ppi_in_254[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 254.

Name	Protocol	Type	Description
ppi_in_255[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_26[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_30[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_40[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_50[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.



Name	Protocol	Type	Description
ppi_in_60[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_70[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_8[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_80[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_9[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 9.
ppi_in_90[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 95.



Name	Protocol	Type	Description
ppi_in_96[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 99.
pvbus_filtermiss_m	PVBus	Master	Passthrough for accesses to pages not used by the GIC IRI.
pvbus_m	PVBus	Master	Memory bus for transactions generated by the GIC.
pvbus_s	PVBus	Slave	Memory bus in.
redistributor_m[256]	GICv3Comms	Master	Input from and output to CPU interface.
reset	Signal	Slave	Resets.
spi_in[988]	Signal	Slave	Shared peripheral interrupts.
wake_request[256]	Signal	Master	Power management outputs.
wire_to_msi_in_0[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 0.
wire_to_msi_in_1[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 1.
wire_to_msi_in_2[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 2.
wire_to_msi_in_3[1024]	Signal	Slave	Wire-to-MSI interrupts for architectural consolidator 3.

## Parameters for GIC\_IRI\_Filter

### A3-affinity-supported

#### Type

bool

#### Default value

0x0

#### Description

Device supports affinity level 3 values that are non-zero.

### ARE-fixed-to-one

#### Type

bool

#### Default value

0x0

#### Description

GICv2 compatibility is not supported and GICD\_CTLR.ARE\_\* is always one.

### CPU-affinities

#### Type

string

**Default value**

0.0.0.0

**Description**

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

**CPU-affinities-file****Type**

string

**Default value**

""

**Description**

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

**DPG-ARE-only****Type**

bool

**Default value**

0x0

**Description**

Limit application of DPG bits to interrupt groups for which ARE=1.

**DPG-bits-implemented****Type**

bool

**Default value**

0x0

**Description**

Enable implementation of interrupt group participation bits or DPG bits in GICR\_CTLR.

**DS-fixed-to-zero****Type**

bool

**Default value**

0x0

**Description**

Enable/disable support of single security state.

**GICD-alias****Type**

int

**Default value**

0x0

**Description**

In GICv2 mode: the base address for a 4k page alias of the first 4k of the Distributor page, in GICv3 mode. the base address of a 64KB page containing message based SPI signalling register aliases(0:Disabled).

**GICD-legacy-registers-as-reserved****Type**

bool

**Default value**

0x0

**Description**

When ARE is RAO/WI, makes superfluous registers in GICD reserved ( including for the purpose of STATUSR updates).

**GICD\_CTLR-DS-1-means-secure-only****Type**

bool

**Default value**

0x0

**Description**

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

**GICD\_ITARGETSR-RAZWI****Type**

bool

**Default value**

0x0

**Description**

If true, the GICD\_ITARGETS registers are RAZ/WI.

**GICD\_PIDR****Type**

int

**Default value**

0x0

**Description**

The value for the GICD\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

**GICD\_TYPER2****Type**

int

**Default value**

0x0

**Description**

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

**GICR-clear-enable-supported****Type**

bool

**Default value**

0x0

**Description**

When true, this sets the value of the RO bit GICR\_CTLR.CES with the value of the parameter allow-LPIEN-clear, making it visible to software.

**GICR-invalidate-registers-implemented****Type**

bool

**Default value**

0x0

**Description**

When true, the registers GICR\_INVLPIR, GICR\_INVALLR and GICR\_SYNCR are implemented.

**GICR\_PIDR****Type**

int

**Default value**

0x0

**Description**

The value for the GICR\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

**GICR\_PROPBASER-read-only****Type**

bool

**Default value**

0x0

**Description**

GICR\_PROPBASER register is read-only.

**GICR\_PROPBASER-reset-value****Type**

int

**Default value**

0x0

**Description**

Value of GICR\_PROPBASER on reset.

**GITS\_BASER0-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER0 register.

**GITS\_BASER0-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER0 register is RAZ/WI.

**GITS\_BASER0-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER0 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER1-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER1 register.

**GITS\_BASER1-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER1 register is RAZ/WI.

**GITS\_BASER1-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER2-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER2 register.

**GITS\_BASER2-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER2 register is RAZ/WI.

**GITS\_BASER2-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER3-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER3 register.

**GITS\_BASER3-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER3 register is RAZ/WI.

**GITS\_BASER3-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER3 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER4-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER4 register.

**GITS\_BASER4-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER4 register is RAZ/WI.

**GITS\_BASER4-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER4 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER5-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER5 register.

**GITS\_BASER5-indirect-RAZ****Type**

bool

**Default value**

0x0



**Description**

Indirect field for GITS\_BASER5 register is RAZ/WI.

**GITS\_BASER5-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER5 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER6-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER6 register.

**GITS\_BASER6-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER6 register is RAZ/WI.

**GITS\_BASER6-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER6 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_BASER7-entry-bytes****Type**

int

**Default value**

0x8

**Description**

Number of bytes required per entry for GITS\_BASER7 register.

**GITS\_BASER7-indirect-RAZ****Type**

bool

**Default value**

0x0

**Description**

Indirect field for GITS\_BASER7 register is RAZ/WI.

**GITS\_BASER7-type****Type**

int

**Default value**

0x0

**Description**

Type field for GITS\_BASER7 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

**GITS\_PIDR****Type**

int

**Default value**

0x0

**Description**

The value for the GITS\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

**ICFGR-PPI-mask****Type**

int

**Default value**

0xaaaaaaaa

**Description**

Mask for writes to ICFGR registers that configure PPIs.

**ICFGR-PPI-reset****Type**

int

**Default value**

0x0

**Description**

Reset value for ICFGR registers that configure PPIs.

**ICFGR-SGI-mask****Type**

int

**Default value**

0x0

**Description**

Mask for writes to ICFGR registers that configure SGIs.

**ICFGR-SGI-reset****Type**

int

**Default value**

0xaaaaaaaa

**Description**

Reset value for ICFGR registers that configure SGIs.

**ICFGR-SPI-mask****Type**

int

**Default value**

0xaaaaaaaa

**Description**

Mask for writes to ICFGR registers that configure SPIs.

**ICFGR-SPI-reset****Type**

int

**Default value**

0x0

**Description**

Reset value for ICFGR registers that configure SPIs.

**ICFGR-rsvd-bit****Type**

bool

**Default value**

0x0

**Description**

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGRn for n>0.

**IGROUP-PPI-mask****Type**

int

**Default value**

0xffff

**Description**

Mask for writes to PPI bits in IGROUP registers.

**IGROUP-PPI-reset****Type**

int

**Default value**

0x0

**Description**

Reset value for SGI bits in IGROUP registers.

**IGROUP-SGI-mask****Type**

int

**Default value**

0xffff

**Description**

Mask for writes to SGI bits in IGROUP registers.

**IGROUP-SGI-reset****Type**

int

**Default value**

0x0

**Description**

Reset value for SGI bits in IGROUP registers.

**IIDR****Type**

int

**Default value**

0x0

**Description**

GICD\_IIDR and GICR\_IIDR value.

**IRI-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits used to represent interrupts IDs in the Distributor and Redistributors, forced to 10 when none of LPIs, extended SPIs or extended PPIs is supported.

**IROUTER-IRM-RAZ-WI****Type**

bool

**Default value**

0x0

**Description**

GICD\_IROUTERn.InterruptRoutingMode is RAZ/WI.

**ITS-BASER-force-page-alignment****Type**

bool

**Default value**

0x1

**Description**

Force alignment of address written to a GITS\_BASER register to the page size configured.

**ITS-ID-bits****Type**

int

**Default value**

0x10

**Description**

Number of interrupt bits supported by ITS.

**ITS-MOVALL-update-collections****Type**

bool

**Default value**

0x0

**Description**

Whether MOVALL command updates the collection entires.

**ITS-TRANSLATE64R****Type**

bool

**Default value**

0x0

**Description**

Add an implementation specific register at 0x10008 supporting 64 bit TRANSLATER (dev[63:32], interrupt[31:0]).

**ITS-cache-invalidate-on-disable****Type**

bool

**Default value**

0x0

**Description**

Sets the RO bit GITS\_TYPER.INV. When true, after the following sequence: 1) GITS\_CTLR.Enabled written 1-->0, 2) GITS\_CTLR.Quiescent observed as 1, 3) GITS\_BASER<n>.Valid written 1-->0, there is no cached information from the ITS memory structure pointed to by GITS\_BASER<n>.

**ITS-collection-ID-bits****Type**

int

**Default value**

0x0

**Description**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0.

**ITS-count****Type**

int

**Default value**

0x0

**Description**

Number of Interrupt Translation Services to be instantiated (0=none).

**ITS-cumulative-collection-tables****Type**

bool

**Default value**

0x1

**Description**

When true, the supported amount of collections is the sum of GITS\_TYPER.HCC and the number of collections supported in memory, otherwise, simply the number supported in memory only. Irrelevant when HCC=0.

**ITS-device-bits****Type**

int

**Default value**

0x10

**Description**

Number of bits supported for ITS device IDs.

**ITS-enable-itt-address-verification****Type**

bool

**Default value**

0x0

**Description**

If true, a transaction will be sent to ITT Address for verification.

**ITS-entry-size****Type**

int

**Default value**

0x8

**Description**

Number of bytes required to store each entry in the ITT tables.

**ITS-hardware-collection-count****Type**

int

**Default value**

0x0

**Description**

Number of hardware collections held exclusively in the ITS.

**ITS-legacy-iidr-typer-offset****Type**

bool

**Default value**

0x0

**Description**

Put the GITS\_IIDR and GITS\_TYPER registers at their older offset of 0x8 and 0x4 respectively.

**ITS-shared-vPE-table****Type**

int

**Default value**

0x0

**Description**

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when has-gicv4.1 is true.

**ITS-threaded-command-queue****Type**

bool

**Default value**

0x1

**Description**

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.



**ITS-use-physical-target-addresses****Type**

bool

**Default value**

0x1

**Description**

Use physical hardware addresses for targets in ITS commands -- must be true for distributed implementations.

**ITS-vmovp-bit****Type**

bool

**Default value**

0x0

**Description**

Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

**ITS0-base****Type**

int

**Default value**

0x0

**Description**

Register base address for ITS0 (automatic if 0).

**ITS1-base****Type**

int

**Default value**

0x0

**Description**

Register base address for ITS1 (automatic if 0).

**ITS2-base****Type**

int

**Default value**

0x0

**Description**

Register base address for ITS2 (automatic if 0).

**ITS3-base****Type**

int

**Default value**

0x0

**Description**

Register base address for ITS3 (automatic if 0).

**LPI-cache-check-data****Type**

bool

**Default value**

0x0

**Description**

Enable Cached LPI data against memory checking when available for cache type.

**LPI-cache-type****Type**

int

**Default value**

0x1

**Description**

Cache type for LPIs, 0:No caching, 1:Full caching.

**MSI\_IIDR****Type**

int

**Default value**

0x0

**Description**

Value returned in MSI\_IIDR registers.

**MSI\_NS-frame0-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 0 registers.

**MSI\_NS-frame0-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

**MSI\_NS-frame0-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

**MSI\_NS-frame1-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 1 registers.

**MSI\_NS-frame1-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

**MSI\_NS-frame1-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

**MSI\_NS-frame2-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 2 registers.

**MSI\_NS-frame2-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

**MSI\_NS-frame2-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

**MSI\_NS-frame3-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 3 registers.

**MSI\_NS-frame3-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

**MSI\_NS-frame3-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

**MSI\_NS-frame4-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 4 registers.

**MSI\_NS-frame4-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

**MSI\_NS-frame4-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

**MSI\_NS-frame5-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 5 registers.

**MSI\_NS-frame5-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

**MSI\_NS-frame5-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

**MSI\_NS-frame6-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 6 registers.

**MSI\_NS-frame6-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

**MSI\_NS-frame6-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

**MSI\_NS-frame7-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for non-secure MSI frame 7 registers.

**MSI\_NS-frame7-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

**MSI\_NS-frame7-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

**MSI\_PIDR****Type**

int

**Default value**

0x0

**Description**

The value for the MSI\_PIDR registers, if non-zero and distributor supports GICv2m. Note: fixed fields (device type etc.) will be overridden in this value.

**MSI\_S-frame0-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 0 registers.

**MSI\_S-frame0-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

**MSI\_S-frame0-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

**MSI\_S-frame1-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 1 registers.

**MSI\_S-frame1-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

**MSI\_S-frame1-min-SPI****Type**

int



**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

**MSI\_S-frame2-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 2 registers.

**MSI\_S-frame2-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

**MSI\_S-frame2-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

**MSI\_S-frame3-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 3 registers.

**MSI\_S-frame3-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

**MSI\_S-frame3-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

**MSI\_S-frame4-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 4 registers.

**MSI\_S-frame4-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

**MSI\_S-frame4-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

**MSI\_S-frame5-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 5 registers.

**MSI\_S-frame5-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

**MSI\_S-frame5-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

**MSI\_S-frame6-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 6 registers.

**MSI\_S-frame6-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

**MSI\_S-frame6-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

**MSI\_S-frame7-base****Type**

int

**Default value**

0x0

**Description**

If non-zero, sets the base address used for secure MSI frame 7 registers.

**MSI\_S-frame7-max-SPI****Type**

int

**Default value**

0x0

**Description**

Maximum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

**MSI\_S-frame7-min-SPI****Type**

int

**Default value**

0x0

**Description**

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

**PA\_SIZE****Type**

int

**Default value**

0x30

**Description**

Number of valid bits in physical address.

**PPI-implemented-mask****Type**

int

**Default value**`0xffff`**Description**

Mask of PPIs that are implemented. One bit per PPI bit 0 == PPI 16 (first PPI). This will affect other masks.

**SPI-count****Type**`int`**Default value**`0xe0`**Description**

Number of SPIs that are implemented.

**SPI-message-based-support****Type**`bool`**Default value**`0x1`**Description**

Distributor supports message based signaling of SPI.

**SPI-unimplemented****Type**`string`**Default value**`""`**Description**

A comma separated list of unimplemented SPIs ranges for sparse SPI definition(for ex: '35, 39-42, 73').

**STATUSR-implemented****Type**`bool`**Default value**`0x1`**Description**

Determines whether the GICR\_STATUSR register is implemented.

**add-output-cpu-wake-request-signal-from-redistributor****Type**

bool

**Default value**

0x0

**Description**

if true, the redistributor will have the output signal `cpu_wake_request` from GIC to DSU and if false, the signals are not added to the redistributor.

**allow-LPIEN-clear****Type**

bool

**Default value**

0x0

**Description**

Allow RW behaviour on `GICR_CTLR.LPIEN` instead of set once.

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged****Type**

bool

**Default value**

0x0

**Description**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the `ISPENDR` register.

**common-lpi-configuration****Type**

int

**Default value**

0x0

**Description**

Describes which re-distributors share (and must be configured with the same) LPI configuration table as described in `GICR_TYPER`( 0:All, 1:A.x.x.x, 2:A.B.x.x, 3:A.B.C.x).

**common-vPE-table-affinity****Type**

string

**Default value**

""

**Description**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s) (where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on). Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when has-gicv4.1 is true.

**consolidators****Type**

string

**Default value**

""

**Description**

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form 'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1, [etc]' (eg '0x3f100000:64:4096, 0x3f200000:64:4224'). The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

**delay-ITS-accesses****Type**

bool

**Default value**

0x1

**Description**

Delay accesses from the ITS until GICR\_SYNCRR is read.

**delay-redistributor-accesses****Type**

bool

**Default value**

0x1

**Description**

Delay memory accesses from the redistributor until GICR\_SYNCRR is read.

**direct-lpi-support****Type**

bool

**Default value**

0x0

**Description**

Enable support for LPI operations through GICR registers.

**enable\_protocol\_checking****Type**

bool

**Default value**

0x0

**Description**

Enable/disable protocol checking at cpu interface.

**enabled****Type**

bool

**Default value**

0x1

**Description**

Enable GICv3 functionality; when false the component is inactive.

**extended-ppi-count****Type**

int

**Default value**

0x0

**Description**

Number of extended PPI supported.

**extended-spi-count****Type**

int

**Default value**

0x0

**Description**

Number of extended SPI supported.

**fixed-routed-spis****Type**

string

**Default value**

""



**Description**

Value of IROUTER[n] register in the form 'n=a.b.c.d, n=\*'. The RM bit of IROUTER is 0 when n=a.b.c.d is used else 1 when n=\* is used. n can be >= 32 and <= 1019.

**gicr-icfgr-extended-count****Type**

int

**Default value**

0x4

**Description**

Number of extended GICR\_ICFGR registers supported.

**gicv2-only****Type**

bool

**Default value**

0x0

**Description**

If true, when using the GICv3 model, pretend to be a GICv2 system.

**group-enables-control-doorbell****Type**

bool

**Default value**

0x0

**Description**

When true, GICR\_VPENDBASER.{VGrp0En,VGrp1En} are cached to allow GIC to check group enables when virtual interrupt targeting this VCPU which is non-resident reaches Redistributor.

**has-gicv4.1****Type**

bool

**Default value**

0x0

**Description**

Enable GICv4.1 functionality; when false the component is inactive.

**has-two-security-states****Type**

bool

**Default value**

0x1

**Description**

If true, has two security states.

**has\_VPENDBASER-dirty-flag-on-load****Type**

bool

**Default value**

0x0

**Description**

GICR\_VPENDBASER.Dirty reflects transient loading state when valid=1.

**has\_mpam****Type**

bool

**Default value**

0x0

**Description**

Enable MPAM support on ITS and RDs.

**has\_nmi****Type**

bool

**Default value**

0x0

**Description**

Enable support for Non-maskable Interrupts (NMIs). (FEAT\_GICv3\_NMI).

**ignore-generate-sgi-when-no-are****Type**

bool

**Default value**

0x0

**Description**

Ignore GenerateSGL packets coming from the CPU interface if both ARE\_S and ARE\_NS are 0.

**individual-doorbell-not-supported****Type**

bool

**Default value**

0x0

**Description**

For IRI with support of virtual interrupt, individual doorbell is not supported when true.

**irouter-default-mask****Type**

string

**Default value**

""

**Description**

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

**irouter-default-reset****Type**

string

**Default value**

""

**Description**

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or '\*'.

**irouter-mask-values****Type**

string

**Default value**

""

**Description**

Mask Value of IROUTER[n] register in the form 'n=a.b.c.d'. n can be  $\geq 32$  and  $\leq 1019$ .

**irouter-reset-values****Type**

string

**Default value**

""

**Description**

Reset Value of IROUTER[n] register in the form 'n=a.b.c.d or n=\*.n can be >= 32 and <= 1019.

**legacy-sgi-enable-rao****Type**

bool

**Default value**

0x0

**Description**

Enables for SGI associated with an ARE=0 regime are RAO/WI.

**local-SEIs****Type**

bool

**Default value**

0x0

**Description**

Generate SEI to signal internal issues.

**local-VSEIs****Type**

bool

**Default value**

0x0

**Description**

Generate VSEI to signal internal issues.

**lockable-SPI-count****Type**

int

**Default value**

0x0

**Description**

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

**monolithic****Type**

bool

**Default value**

0x0

**Description**

Indicate that the implementation is not distributed.

**mpam\_max\_partid****Type**

int

**Default value**

0xffff

**Description**

Maximum valid PARTID.

**mpam\_max\_pmg****Type**

int

**Default value**

0xff

**Description**

Maximum valid PMG.

**non-ARE-core-count****Type**

int

**Default value**

0x8

**Description**

Maximum number of non-ARE cores; normally used to pass the cluster-level NUM\_CORES parameter to the top-level redistributor.

**outer-cacheability-support****Type**

bool

**Default value**

0x0

**Description**

Allow configuration of outer cachability attributes in ITS and Redistributor.

**output\_attributes****Type**

string

**Default value**

"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID,  
ExtendedID[38]=MPAM\_NS"

**Description**

User-defined transform to be applied to bus attributes like MasterID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

**print-memory-map****Type**

bool

**Default value**

0x0

**Description**

Print memory map to stdout.

**priority-bits****Type**

int

**Default value**

0x5

**Description**

Number of implemented priority bits.

**processor-numbers****Type**

string

**Default value**

""

**Description**

Specify processor numbers (as appears in GICR\_TYPER) in the form 0.0.0.0=0,0.0.0.1=1 etc.)  
If not specified, will number processors starting at 0.

**redistributor-threaded-sync****Type**

bool

**Default value**

0x1

**Description**

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

**reg-base****Type**

int

**Default value**

0x2c010000

**Description**

Base for decoding GICv3 registers.

**reg-base-per-redistributor****Type**

string

**Default value**

""

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the reg-base parameter (except that reg-base will still be used for the top-level redistributor). If reg-base-per-redistributor-file is specified, this parameter is ignored.

**reg-base-per-redistributor-file****Type**

string

**Default value**

""

**Description**

Path to file containing the base address for each redistributor in the form '0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000'. All redistributors must be specified and this overrides the reg-base parameter (except that reg-base will still be used for the top-level

redistributor). If this parameter is specified, reg-base-per-redistributor parameter will be ignored even when it is given.

**report-MSI-error-via-statusr****Type**

int

**Default value**

0x0

**Description**

Report MSI error via GITS\_STATUSR. (0:unsupported, 1:report by GITS\_STATUSR, 2:report by GITS\_STATUSR and interrupt as well).

**rme\_default\_mecid\_nonsecure****Type**

int

**Default value**

0x0

**Description**

Default MECID value for NON-SECURE PAS.

**sgi-range-selector-support****Type**

bool

**Default value**

0x0

**Description**

Device has support for the Range Selector feature for SGI.

**single-set-support****Type**

bool

**Default value**

0x0

**Description**

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

**supports-shareability****Type**

bool



**Default value**

0x1

**Description**

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

**trace-speculative-lpi-property-update****Type**

bool

**Default value**

0x0

**Description**

Trace LPI property updates performed on speculative accesses (useful for debugging LPI).

**vPE-table-entry-size-in-doubleword****Type**

int

**Default value**

0x5

**Description**

The size of one entry of a vPE configuration table in double word. The value decremented by one is shown at GICR\_VPROPBASER.Entry\_Size. Current model mandates the minimum entry size to be 4 doublewords. When lower value is given, it is truncated to 4.

**virtual-lpi-support****Type**

bool

**Default value**

0x0

**Description**

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

**virtual-priority-bits****Type**

int

**Default value**

0x5

**Description**

Number of implemented virtual priority bits.

**wakeup-on-reset**

**Type**

bool

**Default value**

0x0

**Description**

Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

**3.10.47 ICS307**

Serially Programmable Clock Source. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1032: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About ICS307**

Use this component to convert the rate of one ClockSignal to another ClockSignal by using configurable multiplier, divider, and scale values. The divider ratio can be set by startup parameters or at runtime by a configuration port. Changes to the input ClockSignal rate and divider ratio are reflected immediately by the output ClockSignal ports.

Three values determine the divisor ratio:

- vdw
- rdw
- od

To calculate the divisor ratio, use:

$$\text{Divisor} = ((\text{rdw}+2) * \text{scale}) / (2 * (\text{vdw}+8))$$

where `scale` is derived from this table indexed by `od`:

**Table 3-1033: od to scale conversion**

od	scale
0	10

od	scale
1	2
2	8
3	4
4	5
5	7
6	3
7	6

The default values of vdw, rdw and od are 4, 6, and 3 to give a default divisor rate of:

$$((6+2) * 4) / (2 * (4+8)) = 4/3$$

### Iris and MTI instances for ICS307

This model has the following Iris instances:

**Table 3-1034: ICS307 Iris instances**

InstanceName	ComponentName
ICS307	ICS307
ICS307.clkdiv_clk1	ClockDivider

This model has the following MTI trace components:

**Table 3-1035: ICS307 MTI instances**

InstanceName	ComponentName
ICS307.clkdiv_clk1	ClockDivider

### Ports for ICS307

**Table 3-1036: Ports**

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Master clock rate.
clk_out_clk1	ClockSignal	Master	Modified clock rate.
clk_out_ref	ClockSignal	Master	Pass through of master clock rate for divider chaining.
configuration	ICS307Configuration	Slave	Configuration port for setting divider ratio dynamically.

### Parameters for ICS307

#### clkdiv\_clk1.div

##### Type

int

##### Default value

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**`clkdiv_clk1.mul`****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**`od`****Type**

int

**Default value**

0x3

**Description**

OD.

**`rdr`****Type**

int

**Default value**

0x6

**Description**

RDR.

**`vdw`****Type**

int

**Default value**

0x4

**Description**

VDW.

### 3.10.48 IDAU

IDAU is a device that provides a Security attribute relating to the address passed to it. For each memory access (data and instruction), the CPU checks the IDAU and sets the security of its transactions based on it. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1037: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### IDAU interface

The Implementation Defined Attribution Unit (IDAU) model uses the `pvr::IDAUSignal` struct to return the Security attributes for the address passed to it.

Unlike the hardware, the CPU Fast Model does not query the IDAU for each access. Communication is at a higher abstraction level to maintain simulation speed.

IDAU interface input signals:

##### **IDAUADDR**

Address of the region

IDAU interface output signals:

##### **IDAUNS**

Non-secure region response

##### **IDAUNSC**

Non-secure-callable region response

##### **IDAUID**

Region number

##### **IDAUIDV**

Region number valid

##### **IDAUNCHK**

Region exempt from attribution check

#### IDAU transaction-level communication protocol

IDAU has the following ports:

- **slave port<PVBUS> pvbus\_s;**  
PVBUS memory-based slave port. Masters can read or write to this port as follows:

- **Read**  
Read returns IDAU region's `pv::IdauRegion` struct (32 byte), containing information about the IDAU region for the requested address. `pv::IdauRegion` contains the start address, end address, `pv::IDAUSignal`, and 8 bytes of padding (to make it 32 byte-aligned).
  - **Write**  
This port only supports 32 byte Write operations to pass in an `pv::IdauRegion` struct for updating an internal IDAU region.
  - **DMI**  
This port adds support for DMI requests and provides a pointer to a `pv::IdauRegion` for the requested address. An 'invalid DMI' call back occurs if the IDAU updates its regions.
- **master port<Value\_64> invalidate\_region;**  
This port is used as a call back to inform masters that the IDAU has updated its region information.



To disable the IDAU, set the `NUM_IDAU_REGION` parameter to zero.

### Iris and MTI instances for IDAU

This model has the following Iris instances:

Table 3-1038: IDAU Iris instances

InstanceName	ComponentName
IDAU.bus_bridge	PVBusBridge

### Ports for IDAU

Table 3-1039: Ports

Name	Protocol	Type	Description
invalidate_region	Value_64	Master	This port is used as a call back to inform masters that the IDAU has updated its region information.
pvbus_s	PVBus	Slave	-

### Parameters for IDAU

**IDAU\_REGION0.BADDR**

**Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region0.

**IDAU\_REGION0.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 0 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION0.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region0 as exempt.

**IDAU\_REGION0.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region0.

**IDAU\_REGION0.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 0 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION1.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region1.

**IDAU\_REGION1.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 1 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION1.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region1 as exempt.

**IDAU\_REGION1.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region1.

**IDAU\_REGION1.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 1 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.



**IDAU\_REGION10.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region10.

**IDAU\_REGION10.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 10 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION10.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region10 as exempt.

**IDAU\_REGION10.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region10.

**IDAU\_REGION10.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 10 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION100.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region100.

**IDAU\_REGION100.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 100 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION100.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region100 as exempt.

**IDAU\_REGION100.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region100.

**IDAU\_REGION100.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 100 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION101.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region101.

**IDAU\_REGION101.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 101 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION101.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region101 as exempt.

**IDAU\_REGION101.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region101.

**IDAU\_REGION101.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 101 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION102.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region102.

**IDAU\_REGION102.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 102 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION102.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region102 as exempt.

**IDAU\_REGION102.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region102.

**IDAU\_REGION102.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 102 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION103.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region103.

**IDAU\_REGION103.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 103 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION103.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region103 as exempt.

**IDAU\_REGION103.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region103.

**IDAU\_REGION103.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 103 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION104.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region104.

**IDAU\_REGION104.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 104 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION104.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region104 as exempt.

**IDAU\_REGION104.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region104.

**IDAU\_REGION104.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 104 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION105.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region105.

**IDAU\_REGION105.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 105 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION105.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region105 as exempt.

**IDAU\_REGION105.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region105.

**IDAU\_REGION105.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 105 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION106.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region106.

**IDAU\_REGION106.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 106 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION106.EXEMPT****Type**

bool



**Default value**

0x0

**Description**

Mark IDAU region106 as exempt.

**IDAU\_REGION106.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region106.

**IDAU\_REGION106.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 106 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION107.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region107.

**IDAU\_REGION107.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 107 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION107.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region107 as exempt.

**IDAU\_REGION107.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region107.

**IDAU\_REGION107.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 107 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION108.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region108.

**IDAU\_REGION108.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 108 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION108.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region108 as exempt.

**IDAU\_REGION108.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region108.

**IDAU\_REGION108.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 108 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION109.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region109.

**IDAU\_REGION109.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 109 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION109.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region109 as exempt.

**IDAU\_REGION109.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region109.

**IDAU\_REGION109.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 109 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION11.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region11.

**IDAU\_REGION11.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 11 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION11.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region11 as exempt.

**IDAU\_REGION11.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region11.

**IDAU\_REGION11.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 11 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION110.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region110.

**IDAU\_REGION110.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 110 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION110.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region110 as exempt.

**IDAU\_REGION110.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region110.

**IDAU\_REGION110.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 110 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION111.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region111.

**IDAU\_REGION111.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 111 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION111.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region111 as exempt.

**IDAU\_REGION111.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region111.

**IDAU\_REGION111.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 111 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION112.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region112.

**IDAU\_REGION112.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 112 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION112.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region112 as exempt.

**IDAU\_REGION112.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region112.

**IDAU\_REGION112.NSC****Type**

bool

**Default value**

0x0



**Description**

Controls if region 112 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION113.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region113.

**IDAU\_REGION113.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 113 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION113.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region113 as exempt.

**IDAU\_REGION113.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region113.

**IDAU\_REGION113.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 113 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION114.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region114.

**IDAU\_REGION114.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 114 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION114.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region114 as exempt.

**IDAU\_REGION114.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region114.

**IDAU\_REGION114.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 114 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION115.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region115.

**IDAU\_REGION115.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 115 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION115.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region115 as exempt.

**IDAU\_REGION115.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region115.

**IDAU\_REGION115.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 115 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION116.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region116.

**IDAU\_REGION116.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 116 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION116.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region116 as exempt.

**IDAU\_REGION116.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region116.

**IDAU\_REGION116.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 116 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION117.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region117.

**IDAU\_REGION117.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 117 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION117.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region117 as exempt.

**IDAU\_REGION117.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region117.

**IDAU\_REGION117.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 117 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION118.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region118.

**IDAU\_REGION118.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 118 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION118.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region118 as exempt.

**IDAU\_REGION118.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region118.

**IDAU\_REGION118.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 118 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION119.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region119.

**IDAU\_REGION119.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 119 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION119.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region119 as exempt.

**IDAU\_REGION119.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region119.

**IDAU\_REGION119.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 119 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION12.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region12.

**IDAU\_REGION12.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 12 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).



**IDAU\_REGION12.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region12 as exempt.

**IDAU\_REGION12.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region12.

**IDAU\_REGION12.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 12 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION120.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region120.

**IDAU\_REGION120.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 120 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION120.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region120 as exempt.

**IDAU\_REGION120.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region120.

**IDAU\_REGION120.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 120 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION121.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region121.

**IDAU\_REGION121.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 121 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION121.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region121 as exempt.

**IDAU\_REGION121.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region121.

**IDAU\_REGION121.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 121 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION122.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region122.

**IDAU\_REGION122.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 122 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION122.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region122 as exempt.

**IDAU\_REGION122.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region122.

**IDAU\_REGION122.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 122 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION123.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region123.

**IDAU\_REGION123.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 123 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION123.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region123 as exempt.

**IDAU\_REGION123.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region123.

**IDAU\_REGION123.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 123 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION124.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region124.

**IDAU\_REGION124.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 124 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION124.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region124 as exempt.

**IDAU\_REGION124.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region124.

**IDAU\_REGION124.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 124 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION125.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region125.

**IDAU\_REGION125.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 125 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION125.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region125 as exempt.

**IDAU\_REGION125.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region125.

**IDAU\_REGION125.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 125 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION126.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region126.

**IDAU\_REGION126.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 126 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION126.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region126 as exempt.

**IDAU\_REGION126.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region126.

**IDAU\_REGION126.NSC****Type**

bool



**Default value**

0x0

**Description**

Controls if region 126 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION127.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region127.

**IDAU\_REGION127.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 127 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION127.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region127 as exempt.

**IDAU\_REGION127.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region127.

**IDAU\_REGION127.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 127 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION128.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region128.

**IDAU\_REGION128.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 128 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION128.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region128 as exempt.

**IDAU\_REGION128.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region128.

**IDAU\_REGION128.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 128 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION129.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region129.

**IDAU\_REGION129.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 129 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION129.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region129 as exempt.

**IDAU\_REGION129.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region129.

**IDAU\_REGION129.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 129 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION13.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region13.

**IDAU\_REGION13.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 13 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION13.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region13 as exempt.

**IDAU\_REGION13.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region13.

**IDAU\_REGION13.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 13 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION130.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region130.

**IDAU\_REGION130.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 130 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION130.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region130 as exempt.

**IDAU\_REGION130.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region130.

**IDAU\_REGION130.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 130 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION131.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region131.

**IDAU\_REGION131.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 131 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION131.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region131 as exempt.

**IDAU\_REGION131.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region131.

**IDAU\_REGION131.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 131 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION132.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region132.

**IDAU\_REGION132.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 132 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION132.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region132 as exempt.

**IDAU\_REGION132.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region132.

**IDAU\_REGION132.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 132 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION133.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region133.

**IDAU\_REGION133.ENABLE****Type**

bool

**Default value**

0x0



**Description**

Controls if region 133 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION133.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region133 as exempt.

**IDAU\_REGION133.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region133.

**IDAU\_REGION133.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 133 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION134.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region134.

**IDAU\_REGION134.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 134 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION134.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region134 as exempt.

**IDAU\_REGION134.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region134.

**IDAU\_REGION134.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 134 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION135.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region135.

**IDAU\_REGION135.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 135 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION135.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region135 as exempt.

**IDAU\_REGION135.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region135.

**IDAU\_REGION135.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 135 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION136.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region136.

**IDAU\_REGION136.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 136 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION136.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region136 as exempt.

**IDAU\_REGION136.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region136.

**IDAU\_REGION136.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 136 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION137.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region137.

**IDAU\_REGION137.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 137 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION137.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region137 as exempt.

**IDAU\_REGION137.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region137.

**IDAU\_REGION137.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 137 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION138.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region138.

**IDAU\_REGION138.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 138 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION138.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region138 as exempt.

**IDAU\_REGION138.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region138.

**IDAU\_REGION138.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 138 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION139.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region139.

**IDAU\_REGION139.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 139 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION139.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region139 as exempt.

**IDAU\_REGION139.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region139.

**IDAU\_REGION139.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 139 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION14.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region14.

**IDAU\_REGION14.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 14 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION14.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region14 as exempt.

**IDAU\_REGION14.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region14.



**IDAU\_REGION14.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 14 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION140.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region140.

**IDAU\_REGION140.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 140 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION140.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region140 as exempt.

**IDAU\_REGION140.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region140.

**IDAU\_REGION140.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 140 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION141.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region141.

**IDAU\_REGION141.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 141 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION141.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region141 as exempt.

**IDAU\_REGION141.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region141.

**IDAU\_REGION141.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 141 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION142.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region142.

**IDAU\_REGION142.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 142 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION142.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region142 as exempt.

**IDAU\_REGION142.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region142.

**IDAU\_REGION142.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 142 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION143.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region143.

**IDAU\_REGION143.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 143 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION143.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region143 as exempt.

**IDAU\_REGION143.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region143.

**IDAU\_REGION143.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 143 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION144.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region144.

**IDAU\_REGION144.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 144 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION144.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region144 as exempt.

**IDAU\_REGION144.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region144.

**IDAU\_REGION144.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 144 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION145.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region145.

**IDAU\_REGION145.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 145 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION145.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region145 as exempt.

**IDAU\_REGION145.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region145.

**IDAU\_REGION145.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 145 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION146.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region146.

**IDAU\_REGION146.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 146 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION146.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region146 as exempt.

**IDAU\_REGION146.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region146.

**IDAU\_REGION146.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 146 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION147.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region147.

**IDAU\_REGION147.ENABLE****Type**

bool



**Default value**

0x0

**Description**

Controls if region 147 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION147.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region147 as exempt.

**IDAU\_REGION147.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region147.

**IDAU\_REGION147.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 147 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION148.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region148.

**IDAU\_REGION148.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 148 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION148.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region148 as exempt.

**IDAU\_REGION148.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region148.

**IDAU\_REGION148.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 148 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION149.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region149.

**IDAU\_REGION149.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 149 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION149.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region149 as exempt.

**IDAU\_REGION149.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region149.

**IDAU\_REGION149.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 149 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION15.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region15.

**IDAU\_REGION15.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 15 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION15.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region15 as exempt.

**IDAU\_REGION15.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region15.

**IDAU\_REGION15.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 15 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION150.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region150.

**IDAU\_REGION150.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 150 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION150.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region150 as exempt.

**IDAU\_REGION150.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region150.

**IDAU\_REGION150.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 150 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION151.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region151.

**IDAU\_REGION151.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 151 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION151.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region151 as exempt.

**IDAU\_REGION151.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region151.

**IDAU\_REGION151.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 151 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION152.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region152.

**IDAU\_REGION152.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 152 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION152.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region152 as exempt.

**IDAU\_REGION152.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region152.

**IDAU\_REGION152.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 152 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION153.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region153.

**IDAU\_REGION153.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 153 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION153.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region153 as exempt.

**IDAU\_REGION153.LADDR****Type**

int

**Default value**

0x0



**Description**

Limit address of IDAU region153.

**IDAU\_REGION153.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 153 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION154.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region154.

**IDAU\_REGION154.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 154 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION154.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region154 as exempt.

**IDAU\_REGION154.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region154.

**IDAU\_REGION154.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 154 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION155.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region155.

**IDAU\_REGION155.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 155 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION155.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region155 as exempt.

**IDAU\_REGION155.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region155.

**IDAU\_REGION155.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 155 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION156.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region156.

**IDAU\_REGION156.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 156 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION156.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region156 as exempt.

**IDAU\_REGION156.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region156.

**IDAU\_REGION156.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 156 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION157.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region157.

**IDAU\_REGION157.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 157 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION157.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region157 as exempt.

**IDAU\_REGION157.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region157.

**IDAU\_REGION157.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 157 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION158.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region158.

**IDAU\_REGION158.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 158 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION158.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region158 as exempt.

**IDAU\_REGION158.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region158.

**IDAU\_REGION158.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 158 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION159.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region159.

**IDAU\_REGION159.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 159 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION159.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region159 as exempt.

**IDAU\_REGION159.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region159.

**IDAU\_REGION159.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 159 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION16.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region16.

**IDAU\_REGION16.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 16 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION16.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region16 as exempt.

**IDAU\_REGION16.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region16.

**IDAU\_REGION16.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 16 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION160.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region160.



**IDAU\_REGION160.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 160 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION160.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region160 as exempt.

**IDAU\_REGION160.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region160.

**IDAU\_REGION160.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 160 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION161.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region161.

**IDAU\_REGION161.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 161 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION161.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region161 as exempt.

**IDAU\_REGION161.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region161.

**IDAU\_REGION161.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 161 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION162.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region162.

**IDAU\_REGION162.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 162 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION162.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region162 as exempt.

**IDAU\_REGION162.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region162.

**IDAU\_REGION162.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 162 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION163.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region163.

**IDAU\_REGION163.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 163 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION163.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region163 as exempt.

**IDAU\_REGION163.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region163.

**IDAU\_REGION163.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 163 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION164.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region164.

**IDAU\_REGION164.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 164 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION164.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region164 as exempt.

**IDAU\_REGION164.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region164.

**IDAU\_REGION164.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 164 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION165.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region165.

**IDAU\_REGION165.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 165 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION165.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region165 as exempt.

**IDAU\_REGION165.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region165.

**IDAU\_REGION165.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 165 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION166.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region166.

**IDAU\_REGION166.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 166 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION166.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region166 as exempt.

**IDAU\_REGION166.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region166.

**IDAU\_REGION166.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 166 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION167.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region167.

**IDAU\_REGION167.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 167 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION167.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region167 as exempt.

**IDAU\_REGION167.LADDR****Type**

int



**Default value**

0x0

**Description**

Limit address of IDAU region167.

**IDAU\_REGION167.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 167 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION168.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region168.

**IDAU\_REGION168.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 168 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION168.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region168 as exempt.

**IDAU\_REGION168.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region168.

**IDAU\_REGION168.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 168 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION169.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region169.

**IDAU\_REGION169.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 169 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION169.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region169 as exempt.

**IDAU\_REGION169.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region169.

**IDAU\_REGION169.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 169 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION17.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region17.

**IDAU\_REGION17.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 17 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION17.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region17 as exempt.

**IDAU\_REGION17.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region17.

**IDAU\_REGION17.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 17 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION170.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region170.

**IDAU\_REGION170.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 170 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION170.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region170 as exempt.

**IDAU\_REGION170.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region170.

**IDAU\_REGION170.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 170 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION171.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region171.

**IDAU\_REGION171.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 171 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION171.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region171 as exempt.

**IDAU\_REGION171.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region171.

**IDAU\_REGION171.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 171 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION172.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region172.

**IDAU\_REGION172.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 172 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION172.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region172 as exempt.

**IDAU\_REGION172.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region172.

**IDAU\_REGION172.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 172 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION173.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region173.

**IDAU\_REGION173.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 173 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION173.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region173 as exempt.

**IDAU\_REGION173.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region173.

**IDAU\_REGION173.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 173 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION174.BADDR****Type**

int

**Default value**

0x0



**Description**

Base address of IDAU region174.

**IDAU\_REGION174.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 174 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION174.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region174 as exempt.

**IDAU\_REGION174.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region174.

**IDAU\_REGION174.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 174 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION175.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region175.

**IDAU\_REGION175.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 175 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION175.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region175 as exempt.

**IDAU\_REGION175.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region175.

**IDAU\_REGION175.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 175 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION176.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region176.

**IDAU\_REGION176.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 176 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION176.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region176 as exempt.

**IDAU\_REGION176.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region176.

**IDAU\_REGION176.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 176 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION177.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region177.

**IDAU\_REGION177.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 177 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION177.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region177 as exempt.

**IDAU\_REGION177.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region177.

**IDAU\_REGION177.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 177 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION178.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region178.

**IDAU\_REGION178.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 178 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION178.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region178 as exempt.

**IDAU\_REGION178.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region178.

**IDAU\_REGION178.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 178 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION179.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region179.

**IDAU\_REGION179.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 179 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION179.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region179 as exempt.

**IDAU\_REGION179.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region179.

**IDAU\_REGION179.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 179 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION18.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region18.

**IDAU\_REGION18.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 18 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION18.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region18 as exempt.

**IDAU\_REGION18.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region18.

**IDAU\_REGION18.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 18 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION180.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region180.

**IDAU\_REGION180.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 180 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION180.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region180 as exempt.



**IDAU\_REGION180.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region180.

**IDAU\_REGION180.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 180 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION181.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region181.

**IDAU\_REGION181.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 181 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION181.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region181 as exempt.

**IDAU\_REGION181.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region181.

**IDAU\_REGION181.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 181 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION182.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region182.

**IDAU\_REGION182.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 182 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION182.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region182 as exempt.

**IDAU\_REGION182.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region182.

**IDAU\_REGION182.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 182 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION183.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region183.

**IDAU\_REGION183.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 183 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION183.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region183 as exempt.

**IDAU\_REGION183.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region183.

**IDAU\_REGION183.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 183 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION184.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region184.

**IDAU\_REGION184.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 184 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION184.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region184 as exempt.

**IDAU\_REGION184.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region184.

**IDAU\_REGION184.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 184 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION185.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region185.

**IDAU\_REGION185.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 185 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION185.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region185 as exempt.

**IDAU\_REGION185.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region185.

**IDAU\_REGION185.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 185 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION186.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region186.

**IDAU\_REGION186.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 186 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION186.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region186 as exempt.

**IDAU\_REGION186.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region186.

**IDAU\_REGION186.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 186 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION187.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region187.

**IDAU\_REGION187.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 187 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION187.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region187 as exempt.

**IDAU\_REGION187.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region187.

**IDAU\_REGION187.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 187 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION188.BADDR****Type**

int



**Default value**

0x0

**Description**

Base address of IDAU region188.

**IDAU\_REGION188.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 188 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION188.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region188 as exempt.

**IDAU\_REGION188.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region188.

**IDAU\_REGION188.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 188 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION189.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region189.

**IDAU\_REGION189.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 189 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION189.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region189 as exempt.

**IDAU\_REGION189.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region189.

**IDAU\_REGION189.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 189 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION19.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region19.

**IDAU\_REGION19.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 19 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION19.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region19 as exempt.

**IDAU\_REGION19.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region19.

**IDAU\_REGION19.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 19 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION190.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region190.

**IDAU\_REGION190.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 190 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION190.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region190 as exempt.

**IDAU\_REGION190.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region190.

**IDAU\_REGION190.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 190 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION191.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region191.

**IDAU\_REGION191.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 191 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION191.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region191 as exempt.

**IDAU\_REGION191.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region191.

**IDAU\_REGION191.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 191 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION192.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region192.

**IDAU\_REGION192.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 192 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION192.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region192 as exempt.

**IDAU\_REGION192.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region192.

**IDAU\_REGION192.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 192 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION193.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region193.

**IDAU\_REGION193.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 193 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION193.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region193 as exempt.

**IDAU\_REGION193.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region193.

**IDAU\_REGION193.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 193 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION194.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region194.

**IDAU\_REGION194.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 194 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION194.EXEMPT****Type**

bool

**Default value**

0x0



**Description**

Mark IDAU region194 as exempt.

**IDAU\_REGION194.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region194.

**IDAU\_REGION194.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 194 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION195.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region195.

**IDAU\_REGION195.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 195 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION195.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region195 as exempt.

**IDAU\_REGION195.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region195.

**IDAU\_REGION195.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 195 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION196.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region196.

**IDAU\_REGION196.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 196 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION196.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region196 as exempt.

**IDAU\_REGION196.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region196.

**IDAU\_REGION196.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 196 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION197.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region197.

**IDAU\_REGION197.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 197 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION197.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region197 as exempt.

**IDAU\_REGION197.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region197.

**IDAU\_REGION197.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 197 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION198.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region198.

**IDAU\_REGION198.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 198 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION198.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region198 as exempt.

**IDAU\_REGION198.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region198.

**IDAU\_REGION198.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 198 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION199.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region199.

**IDAU\_REGION199.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 199 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION199.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region199 as exempt.

**IDAU\_REGION199.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region199.

**IDAU\_REGION199.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 199 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION2.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region2.

**IDAU\_REGION2.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 2 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION2.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region2 as exempt.

**IDAU\_REGION2.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region2.

**IDAU\_REGION2.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 2 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION20.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region20.

**IDAU\_REGION20.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 20 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION20.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region20 as exempt.

**IDAU\_REGION20.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region20.

**IDAU\_REGION20.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 20 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.



**IDAU\_REGION200.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region200.

**IDAU\_REGION200.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 200 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION200.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region200 as exempt.

**IDAU\_REGION200.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region200.

**IDAU\_REGION200.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 200 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION201.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region201.

**IDAU\_REGION201.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 201 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION201.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region201 as exempt.

**IDAU\_REGION201.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region201.

**IDAU\_REGION201.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 201 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION202.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region202.

**IDAU\_REGION202.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 202 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION202.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region202 as exempt.

**IDAU\_REGION202.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region202.

**IDAU\_REGION202.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 202 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION203.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region203.

**IDAU\_REGION203.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 203 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION203.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region203 as exempt.

**IDAU\_REGION203.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region203.

**IDAU\_REGION203.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 203 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION204.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region204.

**IDAU\_REGION204.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 204 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION204.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region204 as exempt.

**IDAU\_REGION204.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region204.

**IDAU\_REGION204.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 204 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION205.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region205.

**IDAU\_REGION205.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 205 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION205.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region205 as exempt.

**IDAU\_REGION205.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region205.

**IDAU\_REGION205.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 205 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION206.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region206.

**IDAU\_REGION206.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 206 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION206.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region206 as exempt.

**IDAU\_REGION206.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region206.

**IDAU\_REGION206.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 206 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION207.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region207.

**IDAU\_REGION207.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 207 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION207.EXEMPT****Type**

bool



**Default value**

0x0

**Description**

Mark IDAU region207 as exempt.

**IDAU\_REGION207.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region207.

**IDAU\_REGION207.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 207 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION208.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region208.

**IDAU\_REGION208.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 208 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION208.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region208 as exempt.

**IDAU\_REGION208.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region208.

**IDAU\_REGION208.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 208 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION209.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region209.

**IDAU\_REGION209.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 209 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION209.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region209 as exempt.

**IDAU\_REGION209.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region209.

**IDAU\_REGION209.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 209 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION21.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region21.

**IDAU\_REGION21.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 21 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION21.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region21 as exempt.

**IDAU\_REGION21.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region21.

**IDAU\_REGION21.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 21 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION210.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region210.

**IDAU\_REGION210.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 210 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION210.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region210 as exempt.

**IDAU\_REGION210.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region210.

**IDAU\_REGION210.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 210 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION211.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region211.

**IDAU\_REGION211.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 211 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION211.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region211 as exempt.

**IDAU\_REGION211.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region211.

**IDAU\_REGION211.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 211 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION212.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region212.

**IDAU\_REGION212.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 212 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION212.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region212 as exempt.

**IDAU\_REGION212.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region212.

**IDAU\_REGION212.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 212 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION213.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region213.

**IDAU\_REGION213.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 213 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION213.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region213 as exempt.

**IDAU\_REGION213.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region213.

**IDAU\_REGION213.NSC****Type**

bool

**Default value**

0x0



**Description**

Controls if region 213 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION214.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region214.

**IDAU\_REGION214.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 214 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION214.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region214 as exempt.

**IDAU\_REGION214.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region214.

**IDAU\_REGION214.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 214 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION215.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region215.

**IDAU\_REGION215.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 215 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION215.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region215 as exempt.

**IDAU\_REGION215.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region215.

**IDAU\_REGION215.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 215 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION216.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region216.

**IDAU\_REGION216.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 216 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION216.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region216 as exempt.

**IDAU\_REGION216.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region216.

**IDAU\_REGION216.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 216 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION217.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region217.

**IDAU\_REGION217.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 217 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION217.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region217 as exempt.

**IDAU\_REGION217.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region217.

**IDAU\_REGION217.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 217 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION218.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region218.

**IDAU\_REGION218.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 218 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION218.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region218 as exempt.

**IDAU\_REGION218.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region218.

**IDAU\_REGION218.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 218 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION219.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region219.

**IDAU\_REGION219.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 219 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION219.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region219 as exempt.

**IDAU\_REGION219.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region219.

**IDAU\_REGION219.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 219 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION22.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region22.

**IDAU\_REGION22.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 22 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION22.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region22 as exempt.

**IDAU\_REGION22.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region22.

**IDAU\_REGION22.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 22 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION220.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region220.

**IDAU\_REGION220.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 220 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).



**IDAU\_REGION220.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region220 as exempt.

**IDAU\_REGION220.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region220.

**IDAU\_REGION220.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 220 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION221.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region221.

**IDAU\_REGION221.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 221 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION221.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region221 as exempt.

**IDAU\_REGION221.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region221.

**IDAU\_REGION221.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 221 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION222.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region222.

**IDAU\_REGION222.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 222 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION222.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region222 as exempt.

**IDAU\_REGION222.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region222.

**IDAU\_REGION222.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 222 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION223.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region223.

**IDAU\_REGION223.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 223 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION223.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region223 as exempt.

**IDAU\_REGION223.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region223.

**IDAU\_REGION223.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 223 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION224.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region224.

**IDAU\_REGION224.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 224 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION224.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region224 as exempt.

**IDAU\_REGION224.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region224.

**IDAU\_REGION224.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 224 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION225.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region225.

**IDAU\_REGION225.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 225 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION225.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region225 as exempt.

**IDAU\_REGION225.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region225.

**IDAU\_REGION225.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 225 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION226.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region226.

**IDAU\_REGION226.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 226 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION226.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region226 as exempt.

**IDAU\_REGION226.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region226.

**IDAU\_REGION226.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 226 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION227.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region227.

**IDAU\_REGION227.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 227 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION227.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region227 as exempt.

**IDAU\_REGION227.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region227.

**IDAU\_REGION227.NSC****Type**

bool



**Default value**

0x0

**Description**

Controls if region 227 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION228.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region228.

**IDAU\_REGION228.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 228 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION228.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region228 as exempt.

**IDAU\_REGION228.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region228.

**IDAU\_REGION228.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 228 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION229.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region229.

**IDAU\_REGION229.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 229 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION229.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region229 as exempt.

**IDAU\_REGION229.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region229.

**IDAU\_REGION229.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 229 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION23.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region23.

**IDAU\_REGION23.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 23 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION23.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region23 as exempt.

**IDAU\_REGION23.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region23.

**IDAU\_REGION23.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 23 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION230.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region230.

**IDAU\_REGION230.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 230 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION230.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region230 as exempt.

**IDAU\_REGION230.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region230.

**IDAU\_REGION230.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 230 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION231.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region231.

**IDAU\_REGION231.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 231 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION231.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region231 as exempt.

**IDAU\_REGION231.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region231.

**IDAU\_REGION231.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 231 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION232.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region232.

**IDAU\_REGION232.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 232 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION232.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region232 as exempt.

**IDAU\_REGION232.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region232.

**IDAU\_REGION232.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 232 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION233.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region233.

**IDAU\_REGION233.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 233 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION233.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region233 as exempt.

**IDAU\_REGION233.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region233.

**IDAU\_REGION233.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 233 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION234.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region234.

**IDAU\_REGION234.ENABLE****Type**

bool

**Default value**

0x0



**Description**

Controls if region 234 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION234.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region234 as exempt.

**IDAU\_REGION234.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region234.

**IDAU\_REGION234.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 234 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION235.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region235.

**IDAU\_REGION235.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 235 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION235.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region235 as exempt.

**IDAU\_REGION235.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region235.

**IDAU\_REGION235.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 235 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION236.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region236.

**IDAU\_REGION236.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 236 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION236.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region236 as exempt.

**IDAU\_REGION236.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region236.

**IDAU\_REGION236.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 236 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION237.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region237.

**IDAU\_REGION237.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 237 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION237.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region237 as exempt.

**IDAU\_REGION237.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region237.

**IDAU\_REGION237.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 237 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION238.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region238.

**IDAU\_REGION238.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 238 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION238.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region238 as exempt.

**IDAU\_REGION238.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region238.

**IDAU\_REGION238.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 238 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION239.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region239.

**IDAU\_REGION239.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 239 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION239.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region239 as exempt.

**IDAU\_REGION239.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region239.

**IDAU\_REGION239.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 239 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION24.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region24.

**IDAU\_REGION24.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 24 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION24.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region24 as exempt.

**IDAU\_REGION24.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region24.

**IDAU\_REGION24.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 24 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION240.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region240.

**IDAU\_REGION240.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 240 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION240.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region240 as exempt.

**IDAU\_REGION240.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region240.



**IDAU\_REGION240.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 240 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION241.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region241.

**IDAU\_REGION241.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 241 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION241.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region241 as exempt.

**IDAU\_REGION241.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region241.

**IDAU\_REGION241.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 241 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION242.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region242.

**IDAU\_REGION242.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 242 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION242.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region242 as exempt.

**IDAU\_REGION242.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region242.

**IDAU\_REGION242.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 242 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION243.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region243.

**IDAU\_REGION243.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 243 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION243.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region243 as exempt.

**IDAU\_REGION243.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region243.

**IDAU\_REGION243.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 243 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION244.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region244.

**IDAU\_REGION244.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 244 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION244.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region244 as exempt.

**IDAU\_REGION244.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region244.

**IDAU\_REGION244.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 244 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION245.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region245.

**IDAU\_REGION245.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 245 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION245.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region245 as exempt.

**IDAU\_REGION245.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region245.

**IDAU\_REGION245.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 245 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION246.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region246.

**IDAU\_REGION246.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 246 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION246.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region246 as exempt.

**IDAU\_REGION246.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region246.

**IDAU\_REGION246.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 246 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION247.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region247.

**IDAU\_REGION247.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 247 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION247.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region247 as exempt.

**IDAU\_REGION247.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region247.

**IDAU\_REGION247.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 247 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION248.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region248.

**IDAU\_REGION248.ENABLE****Type**

bool



**Default value**

0x0

**Description**

Controls if region 248 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION248.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region248 as exempt.

**IDAU\_REGION248.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region248.

**IDAU\_REGION248.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 248 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION249.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region249.

**IDAU\_REGION249.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 249 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION249.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region249 as exempt.

**IDAU\_REGION249.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region249.

**IDAU\_REGION249.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 249 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION25.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region25.

**IDAU\_REGION25.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 25 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION25.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region25 as exempt.

**IDAU\_REGION25.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region25.

**IDAU\_REGION25.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 25 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION250.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region250.

**IDAU\_REGION250.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 250 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION250.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region250 as exempt.

**IDAU\_REGION250.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region250.

**IDAU\_REGION250.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 250 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION251.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region251.

**IDAU\_REGION251.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 251 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION251.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region251 as exempt.

**IDAU\_REGION251.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region251.

**IDAU\_REGION251.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 251 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION252.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region252.

**IDAU\_REGION252.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 252 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION252.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region252 as exempt.

**IDAU\_REGION252.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region252.

**IDAU\_REGION252.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 252 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION253.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region253.

**IDAU\_REGION253.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 253 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION253.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region253 as exempt.

**IDAU\_REGION253.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region253.

**IDAU\_REGION253.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 253 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION254.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region254.

**IDAU\_REGION254.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 254 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION254.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region254 as exempt.

**IDAU\_REGION254.LADDR****Type**

int

**Default value**

0x0



**Description**

Limit address of IDAU region254.

**IDAU\_REGION254.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 254 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION255.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region255.

**IDAU\_REGION255.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 255 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION255.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region255 as exempt.

**IDAU\_REGION255.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region255.

**IDAU\_REGION255.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 255 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION26.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region26.

**IDAU\_REGION26.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 26 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION26.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region26 as exempt.

**IDAU\_REGION26.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region26.

**IDAU\_REGION26.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 26 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION27.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region27.

**IDAU\_REGION27.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 27 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION27.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region27 as exempt.

**IDAU\_REGION27.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region27.

**IDAU\_REGION27.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 27 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION28.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region28.

**IDAU\_REGION28.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 28 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION28.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region28 as exempt.

**IDAU\_REGION28.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region28.

**IDAU\_REGION28.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 28 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION29.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region29.

**IDAU\_REGION29.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 29 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION29.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region29 as exempt.

**IDAU\_REGION29.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region29.

**IDAU\_REGION29.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 29 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION3.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region3.

**IDAU\_REGION3.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 3 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION3.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region3 as exempt.

**IDAU\_REGION3.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region3.

**IDAU\_REGION3.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 3 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION30.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region30.

**IDAU\_REGION30.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 30 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION30.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region30 as exempt.

**IDAU\_REGION30.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region30.

**IDAU\_REGION30.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 30 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION31.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region31.



**IDAU\_REGION31.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 31 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION31.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region31 as exempt.

**IDAU\_REGION31.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region31.

**IDAU\_REGION31.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 31 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION32.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region32.

**IDAU\_REGION32.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 32 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION32.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region32 as exempt.

**IDAU\_REGION32.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region32.

**IDAU\_REGION32.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 32 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION33.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region33.

**IDAU\_REGION33.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 33 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION33.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region33 as exempt.

**IDAU\_REGION33.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region33.

**IDAU\_REGION33.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 33 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION34.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region34.

**IDAU\_REGION34.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 34 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION34.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region34 as exempt.

**IDAU\_REGION34.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region34.

**IDAU\_REGION34.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 34 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION35.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region35.

**IDAU\_REGION35.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 35 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION35.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region35 as exempt.

**IDAU\_REGION35.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region35.

**IDAU\_REGION35.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 35 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION36.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region36.

**IDAU\_REGION36.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 36 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION36.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region36 as exempt.

**IDAU\_REGION36.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region36.

**IDAU\_REGION36.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 36 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION37.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region37.

**IDAU\_REGION37.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 37 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION37.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region37 as exempt.

**IDAU\_REGION37.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region37.

**IDAU\_REGION37.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 37 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION38.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region38.

**IDAU\_REGION38.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 38 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION38.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region38 as exempt.

**IDAU\_REGION38.LADDR****Type**

int



**Default value**

0x0

**Description**

Limit address of IDAU region38.

**IDAU\_REGION38.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 38 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION39.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region39.

**IDAU\_REGION39.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 39 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION39.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region39 as exempt.

**IDAU\_REGION39.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region39.

**IDAU\_REGION39.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 39 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION4.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region4.

**IDAU\_REGION4.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 4 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION4.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region4 as exempt.

**IDAU\_REGION4.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region4.

**IDAU\_REGION4.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 4 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION40.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region40.

**IDAU\_REGION40.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 40 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION40.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region40 as exempt.

**IDAU\_REGION40.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region40.

**IDAU\_REGION40.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 40 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION41.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region41.

**IDAU\_REGION41.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 41 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION41.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region41 as exempt.

**IDAU\_REGION41.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region41.

**IDAU\_REGION41.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 41 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION42.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region42.

**IDAU\_REGION42.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 42 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION42.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region42 as exempt.

**IDAU\_REGION42.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region42.

**IDAU\_REGION42.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 42 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION43.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region43.

**IDAU\_REGION43.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 43 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION43.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region43 as exempt.

**IDAU\_REGION43.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region43.

**IDAU\_REGION43.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 43 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION44.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region44.

**IDAU\_REGION44.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 44 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION44.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region44 as exempt.

**IDAU\_REGION44.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region44.

**IDAU\_REGION44.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 44 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION45.BADDR****Type**

int

**Default value**

0x0



**Description**

Base address of IDAU region45.

**IDAU\_REGION45.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 45 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION45.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region45 as exempt.

**IDAU\_REGION45.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region45.

**IDAU\_REGION45.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 45 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION46.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region46.

**IDAU\_REGION46.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 46 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION46.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region46 as exempt.

**IDAU\_REGION46.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region46.

**IDAU\_REGION46.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 46 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION47.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region47.

**IDAU\_REGION47.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 47 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION47.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region47 as exempt.

**IDAU\_REGION47.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region47.

**IDAU\_REGION47.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 47 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION48.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region48.

**IDAU\_REGION48.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 48 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION48.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region48 as exempt.

**IDAU\_REGION48.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region48.

**IDAU\_REGION48.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 48 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION49.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region49.

**IDAU\_REGION49.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 49 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION49.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region49 as exempt.

**IDAU\_REGION49.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region49.

**IDAU\_REGION49.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 49 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION5.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region5.

**IDAU\_REGION5.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 5 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION5.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region5 as exempt.

**IDAU\_REGION5.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region5.

**IDAU\_REGION5.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 5 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION50.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region50.

**IDAU\_REGION50.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 50 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION50.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region50 as exempt.

**IDAU\_REGION50.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region50.

**IDAU\_REGION50.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 50 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION51.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region51.

**IDAU\_REGION51.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 51 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION51.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region51 as exempt.



**IDAU\_REGION51.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region51.

**IDAU\_REGION51.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 51 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION52.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region52.

**IDAU\_REGION52.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 52 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION52.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region52 as exempt.

**IDAU\_REGION52.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region52.

**IDAU\_REGION52.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 52 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION53.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region53.

**IDAU\_REGION53.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 53 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION53.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region53 as exempt.

**IDAU\_REGION53.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region53.

**IDAU\_REGION53.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 53 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION54.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region54.

**IDAU\_REGION54.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 54 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION54.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region54 as exempt.

**IDAU\_REGION54.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region54.

**IDAU\_REGION54.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 54 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION55.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region55.

**IDAU\_REGION55.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 55 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION55.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region55 as exempt.

**IDAU\_REGION55.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region55.

**IDAU\_REGION55.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 55 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION56.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region56.

**IDAU\_REGION56.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 56 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION56.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region56 as exempt.

**IDAU\_REGION56.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region56.

**IDAU\_REGION56.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 56 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION57.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region57.

**IDAU\_REGION57.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 57 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION57.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region57 as exempt.

**IDAU\_REGION57.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region57.

**IDAU\_REGION57.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 57 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION58.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region58.

**IDAU\_REGION58.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 58 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION58.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region58 as exempt.

**IDAU\_REGION58.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region58.

**IDAU\_REGION58.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 58 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION59.BADDR****Type**

int



**Default value**

0x0

**Description**

Base address of IDAU region59.

**IDAU\_REGION59.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 59 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION59.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region59 as exempt.

**IDAU\_REGION59.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region59.

**IDAU\_REGION59.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 59 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION6.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region6.

**IDAU\_REGION6.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 6 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION6.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region6 as exempt.

**IDAU\_REGION6.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region6.

**IDAU\_REGION6.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 6 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION60.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region60.

**IDAU\_REGION60.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 60 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION60.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region60 as exempt.

**IDAU\_REGION60.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region60.

**IDAU\_REGION60.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 60 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION61.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region61.

**IDAU\_REGION61.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 61 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION61.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region61 as exempt.

**IDAU\_REGION61.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region61.

**IDAU\_REGION61.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 61 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION62.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region62.

**IDAU\_REGION62.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 62 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION62.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region62 as exempt.

**IDAU\_REGION62.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region62.

**IDAU\_REGION62.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 62 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION63.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region63.

**IDAU\_REGION63.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 63 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION63.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region63 as exempt.

**IDAU\_REGION63.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region63.

**IDAU\_REGION63.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 63 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION64.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region64.

**IDAU\_REGION64.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 64 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION64.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region64 as exempt.

**IDAU\_REGION64.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region64.

**IDAU\_REGION64.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 64 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION65.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region65.

**IDAU\_REGION65.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 65 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION65.EXEMPT****Type**

bool

**Default value**

0x0



**Description**

Mark IDAU region65 as exempt.

**IDAU\_REGION65.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region65.

**IDAU\_REGION65.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 65 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION66.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region66.

**IDAU\_REGION66.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 66 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION66.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region66 as exempt.

**IDAU\_REGION66.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region66.

**IDAU\_REGION66.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 66 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION67.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region67.

**IDAU\_REGION67.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 67 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION67.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region67 as exempt.

**IDAU\_REGION67.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region67.

**IDAU\_REGION67.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 67 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION68.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region68.

**IDAU\_REGION68.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 68 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION68.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region68 as exempt.

**IDAU\_REGION68.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region68.

**IDAU\_REGION68.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 68 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION69.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region69.

**IDAU\_REGION69.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 69 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION69.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region69 as exempt.

**IDAU\_REGION69.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region69.

**IDAU\_REGION69.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 69 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION7.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region7.

**IDAU\_REGION7.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 7 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION7.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region7 as exempt.

**IDAU\_REGION7.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region7.

**IDAU\_REGION7.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 7 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION70.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region70.

**IDAU\_REGION70.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 70 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION70.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region70 as exempt.

**IDAU\_REGION70.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region70.

**IDAU\_REGION70.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 70 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION71.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region71.

**IDAU\_REGION71.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 71 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION71.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region71 as exempt.

**IDAU\_REGION71.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region71.

**IDAU\_REGION71.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 71 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.



**IDAU\_REGION72.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region72.

**IDAU\_REGION72.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 72 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION72.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region72 as exempt.

**IDAU\_REGION72.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region72.

**IDAU\_REGION72.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 72 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION73.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region73.

**IDAU\_REGION73.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 73 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION73.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region73 as exempt.

**IDAU\_REGION73.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region73.

**IDAU\_REGION73.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 73 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION74.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region74.

**IDAU\_REGION74.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 74 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION74.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region74 as exempt.

**IDAU\_REGION74.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region74.

**IDAU\_REGION74.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 74 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION75.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region75.

**IDAU\_REGION75.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 75 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION75.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region75 as exempt.

**IDAU\_REGION75.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region75.

**IDAU\_REGION75.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 75 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION76.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region76.

**IDAU\_REGION76.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 76 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION76.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region76 as exempt.

**IDAU\_REGION76.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region76.

**IDAU\_REGION76.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 76 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION77.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region77.

**IDAU\_REGION77.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 77 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION77.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region77 as exempt.

**IDAU\_REGION77.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region77.

**IDAU\_REGION77.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 77 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION78.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region78.

**IDAU\_REGION78.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 78 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION78.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region78 as exempt.

**IDAU\_REGION78.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region78.

**IDAU\_REGION78.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 78 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION79.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region79.

**IDAU\_REGION79.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 79 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION79.EXEMPT****Type**

bool



**Default value**

0x0

**Description**

Mark IDAU region79 as exempt.

**IDAU\_REGION79.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region79.

**IDAU\_REGION79.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 79 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION8.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region8.

**IDAU\_REGION8.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 8 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION8.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region8 as exempt.

**IDAU\_REGION8.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region8.

**IDAU\_REGION8.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 8 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION80.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region80.

**IDAU\_REGION80.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 80 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION80.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region80 as exempt.

**IDAU\_REGION80.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region80.

**IDAU\_REGION80.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 80 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION81.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region81.

**IDAU\_REGION81.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 81 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION81.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region81 as exempt.

**IDAU\_REGION81.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region81.

**IDAU\_REGION81.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 81 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION82.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region82.

**IDAU\_REGION82.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 82 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION82.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region82 as exempt.

**IDAU\_REGION82.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region82.

**IDAU\_REGION82.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 82 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION83.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region83.

**IDAU\_REGION83.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 83 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION83.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region83 as exempt.

**IDAU\_REGION83.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region83.

**IDAU\_REGION83.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 83 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION84.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region84.

**IDAU\_REGION84.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 84 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION84.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region84 as exempt.

**IDAU\_REGION84.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region84.

**IDAU\_REGION84.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 84 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION85.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region85.

**IDAU\_REGION85.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 85 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION85.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region85 as exempt.

**IDAU\_REGION85.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region85.

**IDAU\_REGION85.NSC****Type**

bool

**Default value**

0x0



**Description**

Controls if region 85 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION86.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region86.

**IDAU\_REGION86.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 86 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION86.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region86 as exempt.

**IDAU\_REGION86.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region86.

**IDAU\_REGION86.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 86 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION87.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region87.

**IDAU\_REGION87.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 87 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION87.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region87 as exempt.

**IDAU\_REGION87.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region87.

**IDAU\_REGION87.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 87 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION88.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region88.

**IDAU\_REGION88.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 88 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION88.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region88 as exempt.

**IDAU\_REGION88.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region88.

**IDAU\_REGION88.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 88 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION89.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region89.

**IDAU\_REGION89.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 89 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION89.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region89 as exempt.

**IDAU\_REGION89.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region89.

**IDAU\_REGION89.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 89 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION9.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region9.

**IDAU\_REGION9.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 9 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION9.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region9 as exempt.

**IDAU\_REGION9.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region9.

**IDAU\_REGION9.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 9 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION90.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region90.

**IDAU\_REGION90.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 90 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION90.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region90 as exempt.

**IDAU\_REGION90.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region90.

**IDAU\_REGION90.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 90 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION91.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region91.

**IDAU\_REGION91.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 91 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION91.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region91 as exempt.

**IDAU\_REGION91.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region91.

**IDAU\_REGION91.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 91 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION92.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region92.

**IDAU\_REGION92.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 92 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).



**IDAU\_REGION92.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region92 as exempt.

**IDAU\_REGION92.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region92.

**IDAU\_REGION92.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 92 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION93.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region93.

**IDAU\_REGION93.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 93 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION93.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region93 as exempt.

**IDAU\_REGION93.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region93.

**IDAU\_REGION93.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 93 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION94.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region94.

**IDAU\_REGION94.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 94 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION94.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region94 as exempt.

**IDAU\_REGION94.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region94.

**IDAU\_REGION94.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 94 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION95.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region95.

**IDAU\_REGION95.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 95 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION95.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region95 as exempt.

**IDAU\_REGION95.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region95.

**IDAU\_REGION95.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 95 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION96.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region96.

**IDAU\_REGION96.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 96 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION96.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region96 as exempt.

**IDAU\_REGION96.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region96.

**IDAU\_REGION96.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 96 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION97.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region97.

**IDAU\_REGION97.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 97 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION97.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region97 as exempt.

**IDAU\_REGION97.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region97.

**IDAU\_REGION97.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 97 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION98.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region98.

**IDAU\_REGION98.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 98 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored.  
1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION98.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region98 as exempt.

**IDAU\_REGION98.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region98.

**IDAU\_REGION98.NSC****Type**

bool

**Default value**

0x0

**Description**

Controls if region 98 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**IDAU\_REGION99.BADDR****Type**

int

**Default value**

0x0

**Description**

Base address of IDAU region99.

**IDAU\_REGION99.ENABLE****Type**

bool

**Default value**

0x0

**Description**

Controls if region 99 is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

**IDAU\_REGION99.EXEMPT****Type**

bool

**Default value**

0x0

**Description**

Mark IDAU region99 as exempt.

**IDAU\_REGION99.LADDR****Type**

int

**Default value**

0x0

**Description**

Limit address of IDAU region99.

**IDAU\_REGION99.NSC****Type**

bool



**Default value**

0x0

**Description**

Controls if region 99 is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

**NUM\_IDAU\_REGION****Type**

int

**Default value**

0x0

**Description**

Number of IDAU regions.

### 3.10.49 ILCU

Integraton Layer Control Unit. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1040: IP revisions support**

Revision	Quality level
0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Iris and MTI instances for ILCU**

This model has the following Iris instances:

**Table 3-1041: ILCU Iris instances**

InstanceName	ComponentName
ILCU	ILCU

**Ports for ILCU****Table 3-1042: Ports**

Name	Protocol	Type	Description
apb_control_in	PVBus	Slave	APB3 Subordinate Interface - Access to ILCU registers
clk_div_control_out	Value	Master	3 bits to control the Cryptocell clock divider
crypto_irq_in	Signal	Slave	Interrupt input driven by the Cryptocell
crypto_irq_out	Signal	Master	Interrupt output destined for the processor
ic_trigger_in_ack_out	Signal	Master	Integrity checker trigger acknowledge signal

Name	Protocol	Type	Description
ic_trigger_in_req_in	Signal	Slave	Integrity checker trigger request signal
irq_mux_control_out	Signal	Master	IRQ Mux control bit that internally selects where crypto_irq_in is routed
lcm_seed_lfsr_data_out	Value_64	Master	64 bit seed value for the LFSR interface
lcm_seed_lfsr_valid_out	Signal	Master	Indicates that the lcm_seed_lfsr_data signal is valid
lcs_in	Value	Slave	3 bits giving the lifecycle state from the RSE persistent state interface (PSI)
lcs_valid_in	Signal	Slave	Signal from the RSE PSI indicating whether the lifecycle state is valid
n_coldreset_in	Signal	Slave	ICLU reset in
otpw_otp_is_ready_in	Signal	Slave	Status signal from the OTP wrapper indicating it is ready for commands
secure_gpo_out	Value	Master	16 bits of general purpose output (8 bits programmed + 8 complemented bits driven)
trigger_mux_control_out	Value	Master	2 bits input to the DMA Trigger Mux to internally select the DMA trigger source
trigger_out_ack_in	Signal	Slave	DMA Trigger from one of the muxed sources acknowledge signal
trigger_out_req_out	Signal	Master	DMA Trigger from one of the muxed sources request signal
trng_trigger_out	Signal	Master	DMA Trigger Mux input from the IRQ Mux indicating whether the TRNG complete irq has occurred

## Parameters for ILCU

### LCM\_RNG\_SEED\_NOT\_REQUIRED

#### Type

int

#### Default value

0x0

#### Description

Sets reset value of VALID bit in LCM\_SEED\_CTRL register.

### diagnostics

#### Type

int

#### Default value

0x0

#### Description

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

## 3.10.50 IntegrityChecker

Integrity Checker. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Copyright © 2017–2025 Arm Limited (or its affiliates). All rights reserved.  
Non-Confidential

**Table 3-1043: IP revisions support**

Revision	Quality level
0.5	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Iris and MTI instances for IntegrityChecker

This model has the following Iris instances:

**Table 3-1044: IntegrityChecker Iris instances**

InstanceName	ComponentName
IntegrityChecker	IntegrityChecker
IntegrityChecker.apb	PVBusSlave

This model has the following MTI trace components:

**Table 3-1045: IntegrityChecker MTI instances**

InstanceName	ComponentName
IntegrityChecker.apb	PVBusSlave

### Ports for IntegrityChecker

**Table 3-1046: Ports**

Name	Protocol	Type	Description
apb	PVBus	Slave	APB Subordinate Interface - Access to registers
ic_alarm_out	Signal	Master	Alarm status out signal
ic_interrupt_out	Signal	Master	Interrupt out signal
ic_match_trigger_ack_in	Signal	Slave	IC Match done ack signal
ic_match_trigger_req_out	Signal	Master	IC Match done req signal
pvbus_m	PVBus	Master	To read and write to external memory
reset_in	Signal	Slave	Reset in signal
warm_reset_in	Signal	Slave	Warm Reset in signal

### Parameters for IntegrityChecker

#### ICBC\_RESET\_VALUE

##### Type

int

##### Default value

0x11b

##### Description

ICBC Registers Reset Value.

**ICDL\_CHUNK\_SIZE****Type**

int

**Default value**

0x8

**Description**

ICDL Chunk Size.

**PID0\_RESET\_VALUE****Type**

int

**Default value**

0x0

**Description**

PID0 Registers Reset Value.

**PID1\_RESET\_VALUE****Type**

int

**Default value**

0xb0

**Description**

PID1 Registers Reset Value.

**diagnostics****Type**

int

**Default value**

0x2

**Description**

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

### 3.10.51 IoTSS3\_SystemControl

IoT Subsystem System 3.0 Control registers. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1047: IP revisions support**

Revision	Quality level
3.0	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Iris and MTI instances for IoTSS3\_SystemControl

This model has the following Iris instances:

**Table 3-1048: IoTSS3\_SystemControl Iris instances**

InstanceName	ComponentName
IoTSS3_SystemControl	IoTSS3_SystemControl
IoTSS3_SystemControl.busmaster	PVBusMaster
IoTSS3_SystemControl.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-1049: IoTSS3\_SystemControl MTI instances**

InstanceName	ComponentName
IoTSS3_SystemControl.busmaster	PVBusMaster
IoTSS3_SystemControl.busslave	PVBusSlave

### Ports for IoTSS3\_SystemControl

**Table 3-1050: Ports**

Name	Protocol	Type	Description
busmaster_control	PVTransactionMaster	Master	-
cpu0_lockup_reset_request	Signal	Slave	-
cpu0_warm_reset_request	Signal	Slave	-
cpu0core_ppu_irq	Signal	Slave	-
cpu1_lockup_reset_request	Signal	Slave	-
cpu1_warm_reset_request	Signal	Slave	-
cpu2_lockup_reset_request	Signal	Slave	-
cpu2_warm_reset_request	Signal	Slave	-
cpu3_lockup_reset_request	Signal	Slave	-
cpu3_warm_reset_request	Signal	Slave	-
cpuextnmienable_out[4]	Signal	Master	-
cpuintnmienable_out[4]	Signal	Master	-
cpuwait_out[4]	Signal	Master	-
crypto_ppu_irq	Signal	Slave	-
crypto_warm_reset_request	Signal	Slave	-
dbg_ppu_irq	Signal	Slave	-
dbgen_in	Signal	Slave	-

Name	Protocol	Type	Description
dbgen_out	Signal	Master	-
host_level_reset_request	Signal	Slave	-
initsvtor[4]	Value	Master	-
mgmt_ppu_irq	Signal	Slave	-
niden_in	Signal	Slave	-
niden_out	Signal	Master	-
nonsecure_watchdog_reset_request	Signal	Slave	-
npu0_ppu_irq	Signal	Slave	-
pdc_m_pvb_s	PVBus	Master	-
po_reset	Signal	Master	-
pvbus_s	PVBus	Slave	-
secure_watchdog_reset_request	Signal	Slave	-
slow_clock_watchdog_reset_request	Signal	Slave	-
software_reset_request	Signal	Slave	-
spiden_in	Signal	Slave	-
spiden_out	Signal	Master	-
spniden_in	Signal	Slave	-
spniden_out	Signal	Master	-
subsystem_hardware_reset_request	Signal	Slave	-
sys_ppu_irq	Signal	Slave	-
warm_reset	Signal	Master	-

## Parameters for IoTSS3\_SystemControl

### INITSVTOR\_RST

#### Type

int

#### Default value

0x0

#### Description

Reset int32\_t for INITSVTOR. Should match cpu<i>.INITSVTOR.

### NUMCPU

#### Type

int

#### Default value

0x1

#### Description

Number of Cortex-M CPU cores in the subsystem.

**NUMVMBANK****Type**

int

**Default value**

0x2

**Description**

Number of Volatile Memory Banks.

**SWRESETREQ\_BIT****Type**

int

**Default value**

0x9

**Description**

Bit used for SWRESETREQ.

**cpu0wait****Type**

bool

**Default value**

0x0

**Description**

Whether to hold cpu0 in reset at boot.

**cpu1wait****Type**

bool

**Default value**

0x1

**Description**

Whether to hold cpu1 in reset at boot.

**cpu2wait****Type**

bool

**Default value**

0x1

**Description**

Whether to hold cpu2 in reset at boot.

**cpu3wait**

**Type**

bool

**Default value**

0x1

**Description**

Whether to hold cpu3 in reset at boot.

**diagnostics**

**Type**

int

**Default value**

0x2

**Description**

Diagnostics.

**3.10.52 IoTSS\_AccessControlGate**

IoT Subsystem Access Control Gate. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1051: IP revisions support**

Revision	Quality level
1.0	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Iris and MTI instances for IoTSS\_AccessControlGate**

This model has the following Iris instances:

**Table 3-1052: IoTSS\_AccessControlGate Iris instances**

InstanceName	ComponentName
IoTSS_AccessControlGate	IoTSS_AccessControlGate
IoTSS_AccessControlGate.bus_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-1053: IoTSS\_AccessControlGate MTI instances**

InstanceName	ComponentName
IoTSS_AccessControlGate.bus_mapper	<a href="#">PVBusMapper</a>



## Ports for IoTSS\_AccessControlGate

**Table 3-1054: Ports**

Name	Protocol	Type	Description
ext_gate	Signal	Slave	-
master_ppuhwstat	Value	Slave	-
pdbus_m	PVBus	Master	-
pdbus_s	PVBus	Slave	-
slave_ppuhwstat	Value	Slave	-
wake_request	Signal	Master	-

## Parameters for IoTSS\_AccessControlGate

### enabled

#### Type

bool

#### Default value

0x1

#### Description

Enable the ACG. If disabled, will let all transactions through without side effects.

## 3.10.53 IoTSS\_PeripheralProtectionController

IoT Subsystem Peripheral Protection Controller. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1055: IP revisions support**

Revision	Quality level
1.0	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Iris and MTI instances for IoTSS\_PeripheralProtectionController

This model has the following Iris instances:

**Table 3-1056: IoTSS\_PeripheralProtectionController Iris instances**

InstanceName	ComponentName
IoTSS_PeripheralProtectionController	IoTSS_PeripheralProtectionController
IoTSS_PeripheralProtectionController.bus_mapper0	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper1	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper10	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper11	PVBusMapper

InstanceName	ComponentName
IoTSS_PeripheralProtectionController.bus_mapper12	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper13	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper14	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper15	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper2	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper3	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper4	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper5	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper6	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper7	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper8	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper9	PVBusMapper

This model has the following MTI trace components:

**Table 3-1057: IoTSS\_PeripheralProtectionController MTI instances**

InstanceName	ComponentName
IoTSS_PeripheralProtectionController.bus_mapper0	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper1	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper10	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper11	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper12	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper13	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper14	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper15	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper2	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper3	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper4	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper5	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper6	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper7	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper8	PVBusMapper
IoTSS_PeripheralProtectionController.bus_mapper9	PVBusMapper

### Ports for IoTSS\_PeripheralProtectionController

**Table 3-1058: Ports**

Name	Protocol	Type	Description
cfg_ap	ValueState	Slave	-
cfg_nonsec	ValueState	Slave	-
cfg_sec_resp	ValueState	Slave	-

Name	Protocol	Type	Description
ida_u_invalidate_region	Value_64	Slave	-
ppc_irq	StateSignal	Master	-
pvbus_m[16]	PVBus	Master	-
pvbus_s[16]	PVBus	Slave	-

## Parameters for IoTSS\_PeripheralProtectionController

### DISABLE\_GATING

#### Type

bool

#### Default value

0x0

#### Description

Disable Memory gating logic.

### NONSEC\_MASK

#### Type

int

#### Default value

0x0

#### Description

16-bit wide mask for security checking of ports: 0 = check, 1 = mask.

### PORTx\_ENABLE

#### Type

int

#### Default value

0xffff

#### Description

Enable (1) or disable (0) port x (where x is between 0-15): enable = 1, disable = 0.

### diagnostics

#### Type

int

#### Default value

0x0

#### Description

Diagnostics.

### 3.10.54 KeyManagementUnit

KeyManagementUnit. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1059: IP revisions support**

Revision	Quality level
0.5	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Iris and MTI instances for KeyManagementUnit

This model has the following Iris instances:

**Table 3-1060: KeyManagementUnit Iris instances**

InstanceName	ComponentName
KeyManagementUnit	KeyManagementUnit
KeyManagementUnit.apb	PVBusSlave
KeyManagementUnit.keys_in	PVBusSlave

This model has the following MTI trace components:

**Table 3-1061: KeyManagementUnit MTI instances**

InstanceName	ComponentName
KeyManagementUnit.apb	<a href="#">PVBusSlave</a>
KeyManagementUnit.keys_in	<a href="#">PVBusSlave</a>

#### Ports for KeyManagementUnit

**Table 3-1062: Ports**

Name	Protocol	Type	Description
apb	<a href="#">PVBus</a>	Slave	register access via apb port
hw_keys_in	<a href="#">PVBus</a>	Slave	HW key register access via private apb port - connect to LCM
irq_out	<a href="#">Signal</a>	Master	IRQ signal out
keys_out	<a href="#">PVBus</a>	Master	output keys via port
reset_in	<a href="#">Signal</a>	Slave	Reset signal in

#### Parameters for KeyManagementUnit

##### KMUDKPARV0

##### Type

int

**Default value**

0x0

**Description**

Reset value of the KMUDKPA<0> register. If no hardware key slot is supported (KMUNHWKSLTS is 0), this configuration option must be set to 0x0000\_0000.

**KMUDKPARV1****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUDKPA<1> register. If no hardware key slot is supported (KMUNHWKSLTS is 1), this configuration option must be set to 0x0000\_0000.

**KMUDKPARV2****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUDKPA<2> register. If no hardware key slot is supported (KMUNHWKSLTS is 2), this configuration option must be set to 0x0000\_0000.

**KMUDKPARV3****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUDKPA<3> register. If no hardware key slot is supported (KMUNHWKSLTS is 3), this configuration option must be set to 0x0000\_0000.

**KMUDKPARV4****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUDKPA<4> register. If no hardware key slot is supported (KMUNHWKSLTS is 4), this configuration option must be set to 0x0000\_0000.

**KMUDKPARV5****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUDKPA<5> register. If no hardware key slot is supported (KMUNHWKSLTS is 5), this configuration option must be set to 0x0000\_0000.

**KMUDKPARV6****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUDKPA<6> register. If no hardware key slot is supported (KMUNHWKSLTS is 6), this configuration option must be set to 0x0000\_0000.

**KMUDKPARV7****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUDKPA<7> register. If no hardware key slot is supported (KMUNHWKSLTS is 7), this configuration option must be set to 0x0000\_0000.

**KMUKSCRVO****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUKSC<0> register. If no hardware key slot is supported (KMUNHWKSLTS is 0), this configuration option must be set to 0x0000\_0000.

**KMUKSCRV1****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUKSC<1> register. If no hardware key slot is supported (KMUNHWKSLTS is 1), this configuration option must be set to 0x0000\_0000.

**KMUKSCRV2****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUKSC<2> register. If no hardware key slot is supported (KMUNHWKSLTS is 2), this configuration option must be set to 0x0000\_0000.

**KMUKSCRV3****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUKSC<3> register. If no hardware key slot is supported (KMUNHWKSLTS is 3), this configuration option must be set to 0x0000\_0000.

**KMUKSCRV4****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMUKSC<4> register. If no hardware key slot is supported (KMUNHWKSLTS is 4), this configuration option must be set to 0x0000\_0000.

**KMUKSCRV5****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMKSC<5> register. If no hardware key slot is supported (KMUNHWKSLTS is 5), this configuration option must be set to 0x0000\_0000.

**KMKSCRV6****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMKSC<6> register. If no hardware key slot is supported (KMUNHWKSLTS is 6), this configuration option must be set to 0x0000\_0000.

**KMKSCRV7****Type**

int

**Default value**

0x0

**Description**

Reset value of the KMKSC<7> register. If no hardware key slot is supported (KMUNHWKSLTS is 7), this configuration option must be set to 0x0000\_0000.

**KMUNHWKSLTS****Type**

int

**Default value**

0x7

**Description**

Number of hardware key slots. Supported values: 0..8 (0 = No hardware key slot). The first KMUNHWKSLTS key slots are filled by hardware (typically by LCM) and the rest are software key slots. (Default=7).

**KMUNKS****Type**

int

**Default value**

0x5



**Description**

Selects the number of key slots that the KMU supports. This value is reflected in KMUBC.NKS. 0x1=2, 0x2=4, 0x3=8, 0x4=16, 0x5=32 key slots (Default=0x5).

**KMUTANG**

**Type**

int

**Default value**

0x5

**Description**

Specifies four possible values for the Top Address Nibble Groups of the target crypto devices that the KMU allows software to set in the KMUDKPA<n> registers.

**diagnostics**

**Type**

int

**Default value**

0x2

**Description**

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

**3.10.55 LifeCycleManager**

LifeCycleManager. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1063: IP revisions support**

Revision	Quality level
1.02	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Changes in 11.29.19**

Parameters added:

- DCU\_PERMANENT\_DISABLE\_MASK\_VAL
- VIRGIN\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_CM
- VIRGIN\_DEFAULT\_DCU\_VAL\_LCS\_CM

**Iris and MTI instances for LifeCycleManager**

This model has the following Iris instances:

**Table 3-1064: LifeCycleManager Iris instances**

InstanceName	ComponentName
LifeCycleManager	RSE_LifeCycleManager
LifeCycleManager.apb	PVBusSlave
LifeCycleManager.hw_keys_out	PVBusMaster
LifeCycleManager.nvm_external_out	PVBusMaster

This model has the following MTI trace components:

**Table 3-1065: LifeCycleManager MTI instances**

InstanceName	ComponentName
LifeCycleManager.apb	PVBusSlave
LifeCycleManager.hw_keys_out	PVBusMaster
LifeCycleManager.nvm_external_out	PVBusMaster

### Ports for LifeCycleManager

**Table 3-1066: Ports**

Name	Protocol	Type	Description
apb	PVBus	Slave	APB4 Subordinate Interface - Access to LCM registers and internal OTP/NVM
dcu_force_disable_in	Value	Slave	LCM Force disable DCU_EN signals
hw_keys_out	PVBus	Master	APB manager interface for output of the HW Root of Trust keys
lcm_diagnostic_mode_trig_ack_in	Signal	Slave	LCM diagnostic mode trigger response from DMA.
lcm_diagnostic_mode_trig_req_out	Signal	Master	LCM diagnostic mode trigger request to DMA.
lcm_mission_mode_trig_ack_in	Signal	Slave	LCM mission mode trigger response from DMA.
lcm_mission_mode_trig_req_out	Signal	Master	LCM mission mode trigger request to DMA.
lcm_mission_provisioning_mode_trig_ack_in	Signal	Slave	LCM mission-provisioning mode trigger response from DMA.
lcm_mission_provisioning_mode_trig_req_out	Signal	Master	LCM mission-provisioning mode trigger request to DMA.
lcm_mission_se_mode_trig_ack_in	Signal	Slave	LCM security-enabled life cycle mode trigger response from DMA.
lcm_mission_se_mode_trig_req_out	Signal	Master	LCM security-enabled life cycle mode trigger request to DMA.
lcs	Value	Master	The 3-bit value of the LCS register
lcs_valid_out	Signal	Master	LCM state valid out
nvm_external_out	PVBus	Master	APB manager interface for external NVM (OTP)
psi_dcu_en0_out[96]	Signal	Master	-

Name	Protocol	Type	Description
psi_dcu_en1_out[6]	Signal	Master	Debug Control Enable Values (LCM DCUs). Force disabled Debug Control Enable Values. These signals are driven by the computed fd_dcu_en[5:0] signals..
psi_dcu_gppc_out[16]	Signal	Master	General-Purpose Persistent Configuration. These signals are driven by the LCM gppc[15:0] signals and can be exposed to the SoC.
reset_in	Signal	Slave	LCM reset in
sp_reset_out	Signal	Master	LCM SP_RST_REQ signal out

## Parameters for LifeCycleManager

### DCU\_PERMANENT\_DISABLE\_MASK\_VAL

#### Type

string

#### Default value

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

#### Description

Permanently disables the dcu\_en ports.

### DCU\_SP\_DISABLE\_MASK\_VAL

#### Type

string

#### Default value

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

#### Description

The Secure Provisioning disable mask of the 128-bit DCU signals. Clearing a bit in the mask will force the relevant DCU signal to zero when LCM is in Secure Provisioning Mode (SP\_EN=1).

### DISABLE\_DIRECT\_KEY\_APB\_MASKING

#### Type

int

#### Default value

0x0

#### Description

When set to 1, the Direct Key APB masking feature should be disabled.

### KRTL\_VAL

#### Type

string

#### Default value

0x7bfee0b730dcc241938458dafc2bc784cc15bdabeb84311f59cca8a9a4e8bc35

**Description**

The Krtl value. (256-bit hex).

**OTP\_ADDR\_WIDTH****Type**

int

**Default value**

0xc

**Description**

The OTP bus width (width in bits = OTP\_ADDR\_WIDTH + 2).

**OTP\_MASK\_VAL****Type**

string

**Default value**

0x3aaf46ac792c53b0ffd80fe92129f0e8d80a9048799c5498c19f1a1ce7ddf5b20e2cc8f9456cf4ade2fe

**Description**

This value must be generated using a true random number generator (1536-bit hex).

**OTP\_SIZE\_IN\_BYTES****Type**

int

**Default value**

0x4000

**Description**

The size of the OTP region accessible from the LCM APB-S interface. The maximum size is 60KB.

**PCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_CM****Type**

string

**Default value**

"0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The permanent disable mask of the 128-bit DCU signals when LCS=CM and TP Mode=PCI. Clearing a bit in the mask will force the relevant DCU signal to zero.

**PCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_DM****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The permanent disable mask of the 128-bit DCU signals when LCS=DM and TP Mode=PCI.  
Clearing a bit in the mask will force the relevant DCU signal to zero.

**PCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_RMA****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The permanent disable mask of the 128-bit DCU signals when LCS=RMA and TP Mode=PCI.  
Clearing a bit in the mask will force the relevant DCU signal to zero.

**PCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_SECURE****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The permanent disable mask of the 128-bit DCU signals when LCS=SE and TP Mode=PCI.  
Clearing a bit in the mask will force the relevant DCU signal to zero.

**PCI\_DEFAULT\_DCU\_VAL\_LCS\_CM****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The default reset value of the 128-bit DCU signals when LCS=CM and TP Mode=PCI.

**PCI\_DEFAULT\_DCU\_VAL\_LCS\_DM****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The default reset value of the 128-bit DCU signals when LCS=DM and TP Mode=PCI.

**PCI\_DEFAULT\_DCU\_VAL\_LCS\_RMA****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The default reset value of the 128-bit DCU signals when LCS=RMA and TP Mode=PCI.

**PCI\_DEFAULT\_DCU\_VAL\_LCS\_SECURE****Type**

string

**Default value**

"0x0"

**Description**

The default reset value of the 128-bit DCU signals when LCS=SE and TP Mode=PCI.

**TCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_CM****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The permanent disable mask of the 128-bit DCU signals when LCS=CM and TP Mode=TCI.  
Clearing a bit in the mask will force the relevant DCU signal to zero.

**TCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_DM****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The permanent disable mask of the 128-bit DCU signals when LCS=DM and TP Mode=TCI.  
Clearing a bit in the mask will force the relevant DCU signal to zero.

**TCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_RMA****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The permanent disable mask of the 128-bit DCU signals when LCS=RMA and TP Mode=TCI. Clearing a bit in the mask will force the relevant DCU signal to zero.

**TCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_SECURE****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The permanent disable mask of the 128-bit DCU signals when LCS=SE and TP Mode=TCI. Clearing a bit in the mask will force the relevant DCU signal to zero.

**TCI\_DEFAULT\_DCU\_VAL\_LCS\_CM****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The default reset value of the 128-bit DCU signals when LCS=CM and TP Mode=TCI.

**TCI\_DEFAULT\_DCU\_VAL\_LCS\_DM****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The default reset value of the 128-bit DCU signals when LCS=DM and TP Mode=TCI.

**TCI\_DEFAULT\_DCU\_VAL\_LCS\_RMA****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The default reset value of the 128-bit DCU signals when LCS=RMA and TP Mode=TCI.

**TCI\_DEFAULT\_DCU\_VAL\_LCS\_SECURE****Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The default reset value of the 128-bit DCU signals when LCS=SE and TP Mode=TCI.

**VIRGIN\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_CM**

**Type**

string

**Default value**

"OxFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**Description**

The permanent disable mask of the 128-bit DCU signals when LCS=CM and TP Mode=Virgin. Clearing a bit in the mask will force the relevant DCU signal to zero.

**VIRGIN\_DEFAULT\_DCU\_VAL\_LCS\_CM**

**Type**

string

**Default value**

"OxBFFFFFFFFFFFFFFFFFFFFFFFF55555555"

**Description**

The default reset value of the 128-bit DCU signals when LCS=CM and TP Mode=Virgin.

**diagnostics**

**Type**

int

**Default value**

0x2

**Description**

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

**3.10.56 MHU320AE**

Message Handling Unit 320 AE. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1067: IP revisions support**

Revision	Quality level
MHU320AE	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.



## Iris and MTI instances for MHU320AE

This model has the following Iris instances:

**Table 3-1068: MHU320AE Iris instances**

InstanceName	ComponentName
MHU320AE	MHU320AE
MHU320AE.MHU320AE_FMU	mhu320ae_fmu

This model has the following MTI trace components:

**Table 3-1069: MHU320AE MTI instances**

InstanceName	ComponentName
MHU320AE	MessageHandlingUnitV3

## Ports for MHU320AE

**Table 3-1070: Ports**

Name	Protocol	Type	Description
fmu_cri_out	Signal	Master	-
fmu_eri_out	Signal	Master	-
fmu_reset_in	Signal	Slave	-
pvbus_s_rec	PVBus	Slave	Register access for Receiver/Mailbox
pvbus_s_snd	PVBus	Slave	Register access for Sender/Postbox
rec_combined_irq_out	Signal	Master	All interrupts combined for Receiver/MBX
rec_fast_channel_group_irq_out[32]	Signal	Master	Receiver fast channel group interrupts
rec_fast_channel_irq_out[1024]	Signal	Master	Receiver fast channel interrupts
rec_reset_in	Signal	Slave	Reset signal for Receiver/Mailbox
recv_fmu_pvbus_s	PVBus	Slave	-
send_fmu_pvbus_s	PVBus	Slave	FUSA related FMU signals
snd_combined_irq_out	Signal	Master	All Interrupts combined for Sender/PBX
snd_reset_in	Signal	Slave	Reset signal for Sender/Postbox

## Parameters for MHU320AE

### NUM\_DB\_CH

#### Type

int

#### Default value

0x1

#### Description

Number of DoorBell Channels, default=1.

**NUM\_FAST\_CH****Type**

int

**Default value**

0x1

**Description**

Number of Fast Channels, default=1.

**NUM\_FIFO\_CH****Type**

int

**Default value**

0x1

**Description**

Number of FIFO Channels, default=1.

**auto\_op\_full****Type**

bool

**Default value**

0x0

**Description**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

**diagnostics****Type**

int

**Default value**

0x2

**Description**

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG), default=2.

**fast\_ch\_group\_int\_enable****Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**fast\_ch\_n\_per\_group****Type**

int

**Default value**

0x1

**Description**

Fast Channel num channels per group, default=1.

**fast\_ch\_num\_groups****Type**

int

**Default value**

0x1

**Description**

Fast Channel num of groups, default=1.

**fast\_ch\_word\_size****Type**

int

**Default value**

0x20

**Description**

Fast Channel word size 32bit or 64bit, default=32.

**fifo\_depth****Type**

int

**Default value**

0x4

**Description**

Depth of the FIFO = fifo\_depth + 1.

**fmu\_location****Type**

int

**Default value**

0x2

**Description**

fmu\_location 0-2 (0:SENDER 1:RECEIVER 2:BOTH).

**m16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 16 bit access support to FIFO registers.

**m32ba\_spt****Type**

bool

**Default value**

0x1

**Description**

Mailbox 32 bit access support to FIFO registers.

**m64ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 64 bit access support to FIFO registers.

**m8ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 8 bit access support to FIFO registers.

**monolithic****Type**

bool

**Default value**

0x1

**Description**

Monolithic or Distributed MHU - default: monolithic(true).

**p16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 16 bit access support to FIFO registers.

**p32ba\_spt****Type**

bool

**Default value**

0x1

**Description**

Postbox 32 bit access support to FIFO registers.

**p64ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 64 bit access support to FIFO registers.

**p8ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 8 bit access support to FIFO registers.

### 3.10.57 MMC

Generic Multimedia Card. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1071: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About MMC

This component simulates an SD or SDHC card that is compatible with the [MultiMedia Card Association, MMCA](#) specification version 3.31. The parameters permit configuration of a number of attributes reflected in the CID and CSD registers. You can customize the component further by modifying the supplied MMC model source code directly.

When paired with a PL180\_MCI component, the MMC device model provides emulation of a flexible, persistent storage mechanism.

The MMC component uses a file on the host PC to simulate the storage device. The size of this backing store file determines the reported size of the MMC device. As small sections of this file are paged in by the model, large filesystems can be modeled while making efficient use of host PC memory. The backing store file can contain a partition table and filesystems such as FAT or EXT2.

The image file is a direct bit copy of the contents of an SD card. If the image file that the `p_mmc_file` parameter refers to does not exist, the component behaves as if the card is absent. If the image file is read-only, then the component behaves as if the card is read-only.



Operating system boots often attempt to write to the boot filesystem. They might not work properly if the boot filesystem is on a read-only card.

---

The MMC component does not model card insertion or removal. It models the card having already been inserted at system instantiation time.

You can configure the MMC component to behave as an SDHC card by setting the `card_type` parameter to SDHC. SDHC mode is a model-specific extension, and is not supported by PL180 hardware. It supports filesystems that are larger than 2 GB.

The component supports these commands:

- `MMC_GO_IDLE_STATE`
- `MMC_SEND_OP_COND`
- `MMC_ALL_SEND_CID`
- `MMC_SET_RELATIVE_ADDR`
- `MMC_SET_DSR`
- `MMC_SELDESL_CARD`
- `MMC_SEND_CSD`

- MMC\_SEND\_CID
- MMC\_STOP\_TRANSMISSION
- MMC\_SEND\_STATUS
- MMC\_GO\_INACTIVE\_STATE
- MMC\_READ\_SINGLE\_BLOCK
- MMC\_READ\_MULTIPLE\_BLOCK
- MMC\_SET\_BLOCK\_COUNT
- MMC\_WRITE\_BLOCK
- MMC\_WRITE\_MULTIPLE\_BLOCK
- MMC\_SEND\_EXT\_CSD. This command is supported in SDHC mode only

The block length is 512 bytes. SimGen reports attempts to change it as errors.

The component supports these erase commands (Class 5), but they have no effect on the disk backing storage:

- MMC\_ERASE\_GROUP\_START
- MMC\_ERASE\_GROUP\_END
- MMC\_ERASE

The component does not support these commands:

- MMC\_BUSTEST\_R
- MMC\_BUSTEST\_W

The component does not support stream read and write commands (Classes 1 and 3):

- MMC\_READ\_DAT\_UNTIL\_STOP
- MMC\_WRITE\_DAT\_UNTIL\_STOP
- MMC\_PROGRAM\_CID
- MMC\_PROGRAM\_CSD

The component does not support block oriented write protection commands (Class 6):

- MMC\_SET\_WRITE\_PROT
- MMC\_CLR\_WRITE\_PROT
- MMC\_SEND\_WRITE\_PROT

The component does not support lock card commands (Class 7) or application-specific commands (Class 8):

- MMC\_LOCK\_UNLOCK
- MMC\_APP\_CMD
- MMC\_GEN\_CMD

The component does not support I/O mode commands (Class 9):

- `MMC_FAST_IO`
- `MMC_GO_IRQ_STATE`

The component does not support reserved commands. Using a reserved command sets the `MMC_ST_ER_B_ILLEGAL_COMMAND` bit in the status register of the card. Read this with the `MMC_SEND_STATUS` command.

Use the `p_diagnostics` parameter to select the level of diagnostic output, to help to debug device driver and controller-to-card protocol issues. It supports the following levels:

#### Level 0

None

#### Level 1

Warnings about attempting to change read-only settings.

#### Level 2

Trace of command calls.

#### Level 3

Information about every step in the `MMC_Protocol` interaction.

#### Level 4

Hex dump of every block sent or received.

The registers are not memory mapped. Instead, you access them using relevant MMC commands. The MMC component model makes the registers available through a CADI interface. Modification of these registers through CADI is not recommended, but not prohibited. For example, modifying the card ID (CID) registers can be useful when experimenting with drivers, but direct modification of the `STATUS_REG` register is likely to put the card model into an indeterminate state.

For a full definition of MMC registers, see the MMCA System Summary documentation. Device-specific register information can also be obtained from MMC vendors.

**Table 3-1072: MMC registers**

Name	CADI register number	Description
<code>OCR_REG</code>	<code>0x000</code>	Operating conditions register
<code>CID_REG0</code>	<code>0x004</code>	Card ID bits 127:96
<code>CID_REG1</code>	<code>0x005</code>	Card ID bits 95:64
<code>CID_REG2</code>	<code>0x006</code>	Card ID bits 63:32
<code>CID_REG3</code>	<code>0x007</code>	Card ID bits 31:0
<code>CSD_REG0</code>	<code>0x008</code>	Card specific data bits 127:96
<code>CSD_REG1</code>	<code>0x009</code>	Card specific data bits 95:64
<code>CSD_REG2</code>	<code>0x00a</code>	Card specific data bits 63:32
<code>CSD_REG3</code>	<code>0x00b</code>	Card specific data bit 31:0
<code>RCA_REG</code>	<code>0x00c</code>	Relative card address register
<code>DSR_REG</code>	<code>0x00d</code>	Driver stage register



Name	CADI register number	Description
BLOCKLEN_REG	0x00e	Block length
STATUS_REG	0x00f	Card status
BLOCK_COUNT_REG	0x010	Block count

## Iris and MTI instances for MMC

This model has the following Iris instances:

**Table 3-1073: MMC Iris instances**

InstanceName	ComponentName
MMC	MMC
MMC.timer	ClockTimerThread
MMC.timer.timer	ClockTimerThread64
MMC.timer.timer.thread	SchedulerThread
MMC.timer.timer.thread_event	SchedulerThreadEvent

## Ports for MMC

**Table 3-1074: Ports**

Name	Protocol	Type	Description
card_present	StateSignal	Master	Used to signal whether an MMC image is loaded. It is set if an image is loaded, and is clear if no image is loaded.
clk_in	ClockSignal	Slave	Input clock signal used to drive our 'bus'.
mmc	MMC_Protocol	Slave	The MMC slave port.

## Parameters for MMC

### card\_type

#### Type

string

#### Default value

"SDHC"

#### Description

Card type ('SD' or 'SDHC').

### diagnostics

#### Type

int

#### Default value

0x0

#### Description

Diagnostics level.

**force\_sector\_addressing****Type**

bool

**Default value**

0x0

**Description**

Use sector addressing even on small cards.

**p\_OEMid****Type**

int

**Default value**

0x0

**Description**

Card ID OEM ID.

**p\_fast\_access****Type**

bool

**Default value**

0x1

**Description**

Don't simulate MMC block access delays.

**p\_manid****Type**

int

**Default value**

0x2

**Description**

Card ID Manufacturer ID.

**p\_max\_block\_count****Type**

int

**Default value**

0x80

**Description**

Default maximum block count reg. Default 0x80.

**p\_mmc\_file****Type**

string

**Default value**

"mmc.dat"

**Description**

MMCARD filename.

**p\_prodName****Type**

string

**Default value**

"ARMmmc"

**Description**

Card ID Product Name (6 chars).

**p\_prodRev****Type**

int

**Default value**

0x1

**Description**

Card ID Product Revision.

**p\_sernum****Type**

int

**Default value**

0xca4d0001

**Description**

Card Serial Number.

**support\_unpadded\_images****Type**

bool

**Default value**

0x0

**Description**

Support images that are not a multiple of 512k by padding them to the needed size (SDHC cards only).

3.10.58 MMU\_400

MMU-400 component. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1075: IP revisions support


Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About MMU\_400**

This provides a basic MMU-400 that is configurable. It does not allow arbitrary configuration with respect to how StreamIDs and SSD\_Indexes are derived from the transaction attributes.

Set the `use_label_mapping` parameter to `true` if your upstream devices have labels in the top 16 bits of the transaction MasterID.



Note

The model does not have a concept of AXI-ID, but a transaction can have a MasterID set on it.

Label your upstream components 0...N so that the parameters of this component can map those integers to StreamID and SSD\_Index.

Set `use_label_mapping` to `false` if the StreamID is encoded in the top 16 bits of the MasterID and the bottom 16 bits encode either the SSD\_Index or the SSD state directly, depending on `use_ssd_determination_table`. Typically in hardware, a device emits different AXI-IDs, depending on what it is doing. In the model, MasterIDs are usually not diverse and a device might only emit one MasterID.

If `use_ssd_determination_table` is `true`, the bottom 16 bits of the MasterID encode the SSD\_Index. They must be  $< 2^{\text{ssd\_index\_width}}$ . If it is `false`, they encode the SSD state directly (zero is Secure and nonzero is Non-secure).

This component models all architectural registers that are specified in the Technical Reference Manual (TRM), except that it does not model any of the performance registers, and has the following limitations:

- MMU-400 does not have an SMMU\_STLBSTATUS register because the Secure side is a nominal pass-through. MMU-400 only has stage 2 support and you cannot use stage 2 on the Secure side.
- The SMMU\_NSACR is an alias of the Non-secure SMMU\_ACR. This component models SMMU\_ACR as **RAZ/WI**.
- The \*ACR registers have IMP DEF contents. This component models only the PAGESIZE bit of the SACR, as non-**RAZ/WI**. It models no other IMP DEF registers

### Iris and MTI instances for MMU\_400

This model has the following Iris instances:

**Table 3-1076: MMU\_400 Iris instances**

InstanceName	ComponentName
MMU_400	MMU_400
MMU_400.mmu	MMU_400_BASE
MMU_400.mmu.apb3_control_ns_slv	PVBusSlave
MMU_400.mmu.apb3_control_s_slv	PVBusSlave
MMU_400.mmu.apb4_control_slv	PVBusSlave
MMU_400.mmu.mapper	PVBusMapper
MMU_400.mmu.ptw_dvm_receiver	PVBusMapper
MMU_400.mmu.ptw_master	PVBusMaster
MMU_400.mmu.pvbus_master	PVBusMaster

This model has the following MTI trace components:

**Table 3-1077: MMU\_400 MTI instances**

InstanceName	ComponentName
MMU_400.mmu	MMU_400_BASE
MMU_400.mmu.apb3_control_ns_slv	PVBusSlave
MMU_400.mmu.apb3_control_s_slv	PVBusSlave
MMU_400.mmu.apb4_control_slv	PVBusSlave
MMU_400.mmu.mapper	PVBusMapper
MMU_400.mmu.ptw_dvm_receiver	PVBusMapper
MMU_400.mmu.ptw_master	PVBusMaster
MMU_400.mmu.pvbus_master	PVBusMaster

### Ports for MMU\_400

**Table 3-1078: Ports**

Name	Protocol	Type	Description
apb3_control_ns	PVBus	Slave	APBv3 control port for Non-secure access to the register file. If this port is used do not use the APBv4 port.

Name	Protocol	Type	Description
apb3_control_s	PVBus	Slave	APBv3 control port for Secure access to the register file. If this port is used do not use the APBv4 port.
apb4_control	PVBus	Slave	APBv4 control port for access to the register file. If this port is used do not use the APBv3 ports.
cfg_cttw_in	Signal	Slave	Enables coherent page table walks.
cfgflt_irpt_ns	Signal	Master	Non-secure configuration access fault interrupt. Corresponds to SMMU architectural signal SMMU_NScfglrpt.
cfgflt_irpt_s	Signal	Master	Secure configuration access fault interrupt. Corresponds to SMMU architectural signal SMMU_gCfglrpt.
comb_irpt_ns	Signal	Master	Non-secure combined interrupt.
comb_irpt_s	Signal	Master	Secure combined interrupt.
cxt_irpt_ns	Signal	Master	Non-secure context bank fault.
glblflt_irpt_ns	Signal	Master	Global Non-secure fault interrupt. Corresponds to SMMU architectural signal SMMU_NSGlrpt.
glblflt_irpt_s	Signal	Master	Global Secure fault interrupt. Corresponds to SMMU architectural signal SMMU_glrpt.
priv_internals	MMU_400_Internals	Slave	For internal use only, please do not use.
pvbus_m	PVBus	Master	Downstream port of the MMU, where translated transactions emerge.
pvbus_ptw_m	PVBus	Master	Downstream port for page table walks if configured using the ptw_has_separate_port parameter.
pvbus_s	PVBus	Slave	Upstream port of the MMU. Addresses on the port are in VA/IPA.
reset_in	Signal	Slave	Signal to reset the MMU.

## Parameters for MMU\_400

### **always\_secure\_ssd\_indices**

#### Type

string

#### Default value

""

#### Description

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

### **cfg\_cttw**

#### Type

bool

#### Default value

0x1

#### Description

Perform coherent page table walks.

**dump\_unpredictability\_in\_user\_flags****Type**

bool

**Default value**

0x0

**Description**

Override the user flags to encode unpredictable information (validation only).

**label0\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label0: Read SDD or SSD\_Index.

**label0\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label0: Read Stream ID.

**label0\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label0: Write SDD or SSD\_Index.

**label0\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label0: Write Stream ID.

**label10\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label10: Read SDD or SSD\_Index.

**label10\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label10: Read Stream ID.

**label10\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label10: Write SDD or SSD\_Index.

**label10\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label10: Write Stream ID.

**label11\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label11: Read SDD or SSD\_Index.



**label11\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label11: Read Stream ID.

**label11\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label11: Write SDD or SSD\_Index.

**label11\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label11: Write Stream ID.

**label12\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label12: Read SDD or SSD\_Index.

**label12\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label12: Read Stream ID.

**label12\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label12: Write SDD or SSD\_Index.

**label12\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label12: Write Stream ID.

**label13\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label13: Read SDD or SSD\_Index.

**label13\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label13: Read Stream ID.

**label13\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label13: Write SDD or SSD\_Index.

**label13\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label13: Write Stream ID.

**label14\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label14: Read SDD or SSD\_Index.

**label14\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label14: Read Stream ID.

**label14\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label14: Write SDD or SSD\_Index.

**label14\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label14: Write Stream ID.

**label15\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label15: Read SDD or SSD\_Index.

**label15\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label15: Read Stream ID.

**label15\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label15: Write SDD or SSD\_Index.

**label15\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label15: Write Stream ID.

**label16\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label16: Read SDD or SSD\_Index.

**label16\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label16: Read Stream ID.

**label16\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label16: Write SDD or SSD\_Index.

**label16\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label16: Write Stream ID.

**label17\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label17: Read SDD or SSD\_Index.

**label17\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label17: Read Stream ID.

**label17\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label17: Write SDD or SSD\_Index.

**label17\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label17: Write Stream ID.

**label18\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label18: Read SDD or SSD\_Index.

**label18\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label18: Read Stream ID.

**label18\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label18: Write SDD or SSD\_Index.

**label18\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label18: Write Stream ID.

**label19\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label19: Read SDD or SSD\_Index.

**label19\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label19: Read Stream ID.

**label19\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label19: Write SDD or SSD\_Index.

**label19\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label19: Write Stream ID.

**label1\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label1: Read SDD or SSD\_Index.

**label1\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label1: Read Stream ID.

**label1\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label1: Write SDD or SSD\_Index.

**label1\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label1: Write Stream ID.

**label20\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label20: Read SDD or SSD\_Index.



**label20\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label20: Read Stream ID.

**label20\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label20: Write SDD or SSD\_Index.

**label20\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label20: Write Stream ID.

**label21\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label21: Read SDD or SSD\_Index.

**label21\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label21: Read Stream ID.

**label21\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label21: Write SDD or SSD\_Index.

**label21\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label21: Write Stream ID.

**label22\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label22: Read SDD or SSD\_Index.

**label22\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label22: Read Stream ID.

**label22\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label22: Write SDD or SSD\_Index.

**label22\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label22: Write Stream ID.

**label23\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label23: Read SDD or SSD\_Index.

**label23\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label23: Read Stream ID.

**label23\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label23: Write SDD or SSD\_Index.

**label23\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label23: Write Stream ID.

**label24\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label24: Read SDD or SSD\_Index.

**label24\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label24: Read Stream ID.

**label24\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label24: Write SDD or SSD\_Index.

**label24\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label24: Write Stream ID.

**label25\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label25: Read SDD or SSD\_Index.

**label25\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label25: Read Stream ID.

**label25\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label25: Write SDD or SSD\_Index.

**label25\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label25: Write Stream ID.

**label26\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label26: Read SDD or SSD\_Index.

**label26\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label26: Read Stream ID.

**label26\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label26: Write SDD or SSD\_Index.

**label26\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label26: Write Stream ID.

**label27\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label27: Read SDD or SSD\_Index.

**label27\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label27: Read Stream ID.

**label27\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label27: Write SDD or SSD\_Index.

**label27\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label27: Write Stream ID.

**label28\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label28: Read SDD or SSD\_Index.

**label28\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label28: Read Stream ID.

**label28\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label28: Write SDD or SSD\_Index.

**label28\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label28: Write Stream ID.

**label29\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label29: Read SDD or SSD\_Index.

**label29\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label29: Read Stream ID.

**label29\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label29: Write SDD or SSD\_Index.

**label29\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label29: Write Stream ID.

**label2\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label2: Read SDD or SSD\_Index.



**label12\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label2: Read Stream ID.

**label12\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label2: Write SDD or SSD\_Index.

**label12\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label2: Write Stream ID.

**label130\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label30: Read SDD or SSD\_Index.

**label130\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label30: Read Stream ID.

**label30\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label30: Write SDD or SSD\_Index.

**label30\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label30: Write Stream ID.

**label31\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label31: Read SDD or SSD\_Index.

**label31\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label31: Read Stream ID.

**label31\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label31: Write SDD or SSD\_Index.

**label31\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label31: Write Stream ID.

**label3\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label3: Read SDD or SSD\_Index.

**label3\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label3: Read Stream ID.

**label3\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label3: Write SDD or SSD\_Index.

**label3\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label3: Write Stream ID.

**label14\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label4: Read SDD or SSD\_Index.

**label14\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label4: Read Stream ID.

**label14\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label4: Write SDD or SSD\_Index.

**label14\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label4: Write Stream ID.

**label15\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label5: Read SDD or SSD\_Index.

**label5\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label5: Read Stream ID.

**label5\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label5: Write SDD or SSD\_Index.

**label5\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label5: Write Stream ID.

**label6\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label6: Read SDD or SSD\_Index.

**label6\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label6: Read Stream ID.

**label6\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label6: Write SDD or SSD\_Index.

**label6\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label6: Write Stream ID.

**label7\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label7: Read SDD or SSD\_Index.

**label7\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label7: Read Stream ID.

**label7\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label7: Write SDD or SSD\_Index.

**label17\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label7: Write Stream ID.

**label18\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label8: Read SDD or SSD\_Index.

**label18\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label8: Read Stream ID.

**label18\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label8: Write SDD or SSD\_Index.

**label18\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label8: Write Stream ID.

**label9\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label9: Read SDD or SSD\_Index.

**label9\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label9: Read Stream ID.

**label9\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label9: Write SDD or SSD\_Index.

**label9\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label9: Write Stream ID.

**mmu.seed****Type**

int

**Default value**

0x12345678

**Description**

Seed for SMMU.



**number\_of\_contexts****Type**

int

**Default value**

0x8

**Description**

Number of context banks.

**number\_of\_smrs****Type**

int

**Default value**

0x20

**Description**

Number of stream match registers.

**percent\_tlbstatus\_commits****Type**

int

**Default value**

0xa

**Description**

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

**prefetch\_only\_requests****Type**

int

**Default value**

0x0

**Description**

Handle prefetch-only requests by:- 0 -- deny them 1 -- use debug table walks and TLB entries 2 -- treat them as normal transactions (dangerous).

**programmable\_non\_secure\_by\_default\_ssd\_indices****Type**

string

**Default value**

""

**Description**

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

**`programmable_secure_by_default_ssd_indices`****Type**

string

**Default value**

""

**Description**

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

**`ptw_has_separate_port`****Type**

bool

**Default value**

0x1

**Description**

Page Table Walks use pvbus\_ptw\_m.

**`pvbus_m_is_ace_lite`****Type**

bool

**Default value**

0x1

**Description**

Is pvbus\_m (the downstream port that translated transaction exit) ACE-Lite.

**`pvbus_ptw_m_is_ace_lite`****Type**

bool

**Default value**

0x1

**Description**

Is pvbus\_ptw\_m (the downstream port that is used for walks if ptw\_has\_separate\_port is true) ACE-Lite.

**`stream_id_width`****Type**

int

**Default value**

0x6

**Description**

StreamID bit width.

**tlb\_depth**

**Type**

int

**Default value**

0x40

**Description**

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

**use\_label\_mapping**

**Type**

bool

**Default value**

0x1

**Description**

Use label mapping.

**use\_ssd\_determination\_table**

**Type**

bool

**Default value**

0x1

**Description**

Use SSD Determination Table.

**3.10.59 MMU\_400\_BASE**

MMU-400 base component. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1079: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Iris and MTI instances for MMU\_400\_BASE

This model has the following Iris instances:

**Table 3-1080: MMU\_400\_BASE Iris instances**

InstanceName	ComponentName
MMU_400_BASE	MMU_400_BASE
MMU_400_BASE.apb3_control_ns_slv	PVBusSlave
MMU_400_BASE.apb3_control_s_slv	PVBusSlave
MMU_400_BASE.apb4_control_slv	PVBusSlave
MMU_400_BASE.mapper	PVBusMapper
MMU_400_BASE.ptw_dvm_receiver	PVBusMapper
MMU_400_BASE.ptw_master	PVBusMaster
MMU_400_BASE.pvbus_master	PVBusMaster

This model has the following MTI trace components:

**Table 3-1081: MMU\_400\_BASE MTI instances**

InstanceName	ComponentName
MMU_400_BASE	MMU_400_BASE
MMU_400_BASE.apb3_control_ns_slv	PVBusSlave
MMU_400_BASE.apb3_control_s_slv	PVBusSlave
MMU_400_BASE.apb4_control_slv	PVBusSlave
MMU_400_BASE.mapper	PVBusMapper
MMU_400_BASE.ptw_dvm_receiver	PVBusMapper
MMU_400_BASE.ptw_master	PVBusMaster
MMU_400_BASE.pvbus_master	PVBusMaster

## Ports for MMU\_400\_BASE

**Table 3-1082: Ports**

Name	Protocol	Type	Description
apb3_control_ns	PVBus	Slave	If the device has been configured with APB3 control ports then this is used to address the register file with non-secure accesses. If this is the case then the apb4_control port should not be used.
apb3_control_s	PVBus	Slave	If the device has been configured with APB3 control ports then this is used to address the register file with secure accesses. If this is the case then the apb4_control port should not be used.
apb4_control	PVBus	Slave	If the device has been configured with APB4 control ports then this port is used -- it carries the security world with the transaction itself. If this is the case then the apb3_control_s and apb3_control_ns should not be used.

Name	Protocol	Type	Description
cfg_cttw_in	Signal	Slave	The SoC supports coherent page walks, this is meant to be sampled at reset. However, in practice the model has to prevent the race condition between cfg_cttw being asserted at the same 'cycle' as negedge reset. Thus we actually only sample the signal on the first transaction to the SMMU or the first transition on this signal after reset. Thus in the model, we require that cfg_cttw be held for at least this period of time.
cfgflt_irpt_ns	Signal	Master	Non-secure Configuration Access Fault Interrupt In the SMMU Architecture this is called SMMU_NSgCflrpt.
cfgflt_irpt_s	Signal	Master	Secure Configuration Access Fault Interrupt In the SMMU Architecture this is called SMMU_gCflrpt.
comb_irpt_ns	Signal	Master	"Non-secure combined interrupt" (cfgflt_irpt_ns   glblflt_irpt_ns   cxt_irpt_ns)?
comb_irpt_s	Signal	Master	"Secure combined interrupt"
cxt_irpt_ns	Signal	Master	Non-secure context bank fault NOTE that there is only one context bank fault, despite there being potentially 8 contexts. As we are HW stage 2 only then we can't have any banks configured as secure (well if we do then we generate a global fault).
glblflt_irpt_ns	Signal	Master	Global non-secure Fault Interrupt In the SMMU Architecture this is called SMMU_NSglrpt.
glblflt_irpt_s	Signal	Master	Global secure Fault Interrupt In the SMMU Architecture this is called SMMU_glrpt.
identify	MMU_400_BASE_IDENTIFY	Master	This port is a special model port that is used to take a transaction and map it to an SSD/SSD_Index and StreamID.
priv_internals	MMU_400_Internals	Slave	For internal use only, please do not use.
pvbus_m	PVBus	Master	This downstream port is where the translated accesses from pvbus_s emerge. If page walks are configured to come out of this port, then they will come out with the with the same attributes as described for pvbus_ptw_m.
pvbus_ptw_m	PVBus	Master	This downstream port is where page table walk accesses come from. This is only used if configured to use a separate page table walk port. The MMU-400 will only obey DVM messages if configured to use this port. The page walks come out of this port with the following master_id and user_flags. master_id : 0xFFFFFFFF The user flags : user_flags[7:0] stage 1 context_id (or 0xFF if stage2 only) user_flags[15:8] stage 2 context_id (or 0xFF if stage 1 with stage 2 bypass) user_flags[18:16] stage 1 level user_flags[21:19] stage 2 level user_flags[23:22] stage 1 descriptor encoding (0=v7s, 1=v7l, 2=v8l, 3=none) user_flags[25:24] stage 2 descriptor encoding (0=v7s, 1=v7l, 2=v8l, 3=none) user_flags[31:30] adomain of the transaction NOTE that if the walk is being done for a stage 1 page walk descriptor fetch then the stage 1 level field will indicate that level. If the walk is being done for a stage 2 descriptor fetch, then the stage 2 level field will show that level. If the context-id for a stage is not valid (0xFF) then the 'level' information is 0x7.
pvbus_s	PVBus	Slave	This port is the upstream port of the device, addresses on the port are in the VA/IPA
reset_in	Signal	Slave	The reset pin.

## Parameters for MMU\_400\_BASE

### **always\_secure\_ssd\_indices**

**Type**

string

**Default value**

""

**Description**

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

### **cfg\_cttw**

**Type**

bool

**Default value**

0x1

**Description**

Perform coherent page table walks.

### **dump\_unpredictability\_in\_user\_flags**

**Type**

bool

**Default value**

0x0

**Description**

Override the user flags to encode unpredictable information (validation only).

### **number\_of\_contexts**

**Type**

int

**Default value**

0x8

**Description**

Number of context banks.

### **number\_of\_smrs**

**Type**

int

**Default value**

0x10

**Description**

Number of stream match registers.

**percent\_tlbstatus\_commits****Type**

int

**Default value**

0xa

**Description**

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

**prefetch\_only\_requests****Type**

int

**Default value**

0x0

**Description**

Handle prefetch-only requests by:- 0 -- deny them 1 -- use debug table walks and TLB entries 2 -- treat them as normal transactions (dangerous).

**programmable\_non\_secure\_by\_default\_ssd\_indices****Type**

string

**Default value**

""

**Description**

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

**programmable\_secure\_by\_default\_ssd\_indices****Type**

string

**Default value**

""

**Description**

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

**ptw\_has\_separate\_port****Type**

bool

**Default value**

0x1

**Description**

Page Table Walks use pvbus\_ptw\_m.

**pvbus\_m\_is\_ace\_lite****Type**

bool

**Default value**

0x1

**Description**

Is pvbus\_m (the downstream port that translated transaction exit) ACE-Lite.

**pvbus\_ptw\_m\_is\_ace\_lite****Type**

bool

**Default value**

0x1

**Description**

Is pvbus\_ptw\_m (the downstream port that is used for walks if ptw\_has\_separate\_port is true) ACE-Lite.

**seed****Type**

int

**Default value**

0x12345678

**Description**

Seed for SMMU.

**stream\_id\_width****Type**

int

**Default value**

0x6

**Description**

StreamID bit width.



**tlb\_depth**  
**Type**  
int  
**Default value**  
0x40  
**Description**  
TLB Depth (0 means 10000). The model will perform best with more TLB entries.

**use\_ssd\_determination\_table**  
**Type**  
bool  
**Default value**  
0x1  
**Description**  
Use SSD Determination Table.

### 3.10.60 MMU\_500

MMU-500 component. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1083: IP revisions support

Revision	Quality level
r2p4	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About MMU\_500

This is a model of a basic MMU-500. Set the version using the `version` parameter.

You cannot arbitrarily configure how you derive StreamIDs and SSD\_Indexes from the transaction attributes.

This component has two label modes which you select using the parameter `use_label_mapping`:

- Set `use_label_mapping` to `true` if your upstream devices have labels in the top 16 bits of the transaction MasterID.



The model does not have a concept of AXI-ID, but a transaction can have a MasterID set on it.

Label your upstream components 0...N so that the parameters of this component can map those integers to StreamID and SSD\_Index.

- Set `use_label_mapping` to `false` if the StreamID is encoded in the top 16 bits of the MasterID and the bottom 16 bits encode either the SSD\_Index or the SSD state directly, depending on `use_ssd_determination_table`:
  - If `use_ssd_determination_table` is `true`, the bottom 16 bits of the MasterID encode the SSD\_Index. They must be  $< 2^{\text{ssd\_index\_width}}$ .
  - If `use_ssd_determination_table` is `false`, the bottom 16 bits of the MasterID encode the SSD state directly, where zero is Secure and nonzero is Non-secure.

Typically in hardware, a device emits different AXI-IDs, depending on what it is doing. In the model, MasterIDs are usually not diverse and a device might only emit one MasterID.

This component models the registers as follows:

- It models all architectural registers that the Technical Reference Manual (TRM) specifies, except that it does not model any of the performance registers.
- Unlike the MMU-400, MMU-500 does have an SMMU\_STLBGSTATUS register because it has stage 1 and stage 2 support.
- The SMMU\_NSACR is an alias of the Non-secure SMMU\_ACR. This component models SMMU\_ACR as **RAZ/WI**.
- The \*ACR registers have IMP DEF contents. This component models only the PAGESIZE bit of the SACR, as non-**RAZ/WI**. It models no other IMP DEF registers.

## Iris and MTI instances for MMU\_500

This model has the following Iris instances:

**Table 3-1084: MMU\_500 Iris instances**

InstanceName	ComponentName
MMU_500	MMU_500
MMU_500.mmu	MMU_500_BASE
MMU_500.mmu.mapper0	PVBusMapper
MMU_500.mmu.mapper1	PVBusMapper
MMU_500.mmu.mapper10	PVBusMapper
MMU_500.mmu.mapper11	PVBusMapper
MMU_500.mmu.mapper12	PVBusMapper
MMU_500.mmu.mapper13	PVBusMapper
MMU_500.mmu.mapper14	PVBusMapper
MMU_500.mmu.mapper15	PVBusMapper
MMU_500.mmu.mapper16	PVBusMapper
MMU_500.mmu.mapper17	PVBusMapper
MMU_500.mmu.mapper18	PVBusMapper
MMU_500.mmu.mapper19	PVBusMapper

InstanceName	ComponentName
MMU_500.mmu.mapper2	PVBusMapper
MMU_500.mmu.mapper20	PVBusMapper
MMU_500.mmu.mapper21	PVBusMapper
MMU_500.mmu.mapper22	PVBusMapper
MMU_500.mmu.mapper23	PVBusMapper
MMU_500.mmu.mapper24	PVBusMapper
MMU_500.mmu.mapper25	PVBusMapper
MMU_500.mmu.mapper26	PVBusMapper
MMU_500.mmu.mapper27	PVBusMapper
MMU_500.mmu.mapper28	PVBusMapper
MMU_500.mmu.mapper29	PVBusMapper
MMU_500.mmu.mapper3	PVBusMapper
MMU_500.mmu.mapper30	PVBusMapper
MMU_500.mmu.mapper31	PVBusMapper
MMU_500.mmu.mapper4	PVBusMapper
MMU_500.mmu.mapper5	PVBusMapper
MMU_500.mmu.mapper6	PVBusMapper
MMU_500.mmu.mapper7	PVBusMapper
MMU_500.mmu.mapper8	PVBusMapper
MMU_500.mmu.mapper9	PVBusMapper
MMU_500.mmu.ptw_dvm_receiver	PVBusMapper
MMU_500.mmu.ptw_master	PVBusMaster
MMU_500.mmu.pvbus_control_s_slv	PVBusSlave
MMU_500.mmu.pvbus_master	PVBusMaster
MMU_500.mmu.pvbusmaster0	PVBusMaster
MMU_500.mmu.pvbusmaster1	PVBusMaster
MMU_500.mmu.pvbusmaster10	PVBusMaster
MMU_500.mmu.pvbusmaster11	PVBusMaster
MMU_500.mmu.pvbusmaster12	PVBusMaster
MMU_500.mmu.pvbusmaster13	PVBusMaster
MMU_500.mmu.pvbusmaster14	PVBusMaster
MMU_500.mmu.pvbusmaster15	PVBusMaster
MMU_500.mmu.pvbusmaster16	PVBusMaster
MMU_500.mmu.pvbusmaster17	PVBusMaster
MMU_500.mmu.pvbusmaster18	PVBusMaster
MMU_500.mmu.pvbusmaster19	PVBusMaster
MMU_500.mmu.pvbusmaster2	PVBusMaster
MMU_500.mmu.pvbusmaster20	PVBusMaster
MMU_500.mmu.pvbusmaster21	PVBusMaster

InstanceName	ComponentName
MMU_500.mmu.pvbusmaster22	PVBusMaster
MMU_500.mmu.pvbusmaster23	PVBusMaster
MMU_500.mmu.pvbusmaster24	PVBusMaster
MMU_500.mmu.pvbusmaster25	PVBusMaster
MMU_500.mmu.pvbusmaster26	PVBusMaster
MMU_500.mmu.pvbusmaster27	PVBusMaster
MMU_500.mmu.pvbusmaster28	PVBusMaster
MMU_500.mmu.pvbusmaster29	PVBusMaster
MMU_500.mmu.pvbusmaster3	PVBusMaster
MMU_500.mmu.pvbusmaster30	PVBusMaster
MMU_500.mmu.pvbusmaster31	PVBusMaster
MMU_500.mmu.pvbusmaster4	PVBusMaster
MMU_500.mmu.pvbusmaster5	PVBusMaster
MMU_500.mmu.pvbusmaster6	PVBusMaster
MMU_500.mmu.pvbusmaster7	PVBusMaster
MMU_500.mmu.pvbusmaster8	PVBusMaster
MMU_500.mmu.pvbusmaster9	PVBusMaster
MMU_500.mmu.pvbusslave0	PVBusSlave
MMU_500.mmu.pvbusslave1	PVBusSlave
MMU_500.mmu.pvbusslave10	PVBusSlave
MMU_500.mmu.pvbusslave11	PVBusSlave
MMU_500.mmu.pvbusslave12	PVBusSlave
MMU_500.mmu.pvbusslave13	PVBusSlave
MMU_500.mmu.pvbusslave14	PVBusSlave
MMU_500.mmu.pvbusslave15	PVBusSlave
MMU_500.mmu.pvbusslave16	PVBusSlave
MMU_500.mmu.pvbusslave17	PVBusSlave
MMU_500.mmu.pvbusslave18	PVBusSlave
MMU_500.mmu.pvbusslave19	PVBusSlave
MMU_500.mmu.pvbusslave2	PVBusSlave
MMU_500.mmu.pvbusslave20	PVBusSlave
MMU_500.mmu.pvbusslave21	PVBusSlave
MMU_500.mmu.pvbusslave22	PVBusSlave
MMU_500.mmu.pvbusslave23	PVBusSlave
MMU_500.mmu.pvbusslave24	PVBusSlave
MMU_500.mmu.pvbusslave25	PVBusSlave
MMU_500.mmu.pvbusslave26	PVBusSlave
MMU_500.mmu.pvbusslave27	PVBusSlave
MMU_500.mmu.pvbusslave28	PVBusSlave

InstanceName	ComponentName
MMU_500.mmu.pvbusslave29	PVBusSlave
MMU_500.mmu.pvbusslave3	PVBusSlave
MMU_500.mmu.pvbusslave30	PVBusSlave
MMU_500.mmu.pvbusslave31	PVBusSlave
MMU_500.mmu.pvbusslave4	PVBusSlave
MMU_500.mmu.pvbusslave5	PVBusSlave
MMU_500.mmu.pvbusslave6	PVBusSlave
MMU_500.mmu.pvbusslave7	PVBusSlave
MMU_500.mmu.pvbusslave8	PVBusSlave
MMU_500.mmu.pvbusslave9	PVBusSlave

This model has the following MTI trace components:

**Table 3-1085: MMU\_500 MTI instances**

InstanceName	ComponentName
MMU_500.mmu	MMU_500_BASE
MMU_500.mmu.mapper0	PVBusMapper
MMU_500.mmu.mapper1	PVBusMapper
MMU_500.mmu.mapper10	PVBusMapper
MMU_500.mmu.mapper11	PVBusMapper
MMU_500.mmu.mapper12	PVBusMapper
MMU_500.mmu.mapper13	PVBusMapper
MMU_500.mmu.mapper14	PVBusMapper
MMU_500.mmu.mapper15	PVBusMapper
MMU_500.mmu.mapper16	PVBusMapper
MMU_500.mmu.mapper17	PVBusMapper
MMU_500.mmu.mapper18	PVBusMapper
MMU_500.mmu.mapper19	PVBusMapper
MMU_500.mmu.mapper2	PVBusMapper
MMU_500.mmu.mapper20	PVBusMapper
MMU_500.mmu.mapper21	PVBusMapper
MMU_500.mmu.mapper22	PVBusMapper
MMU_500.mmu.mapper23	PVBusMapper
MMU_500.mmu.mapper24	PVBusMapper
MMU_500.mmu.mapper25	PVBusMapper
MMU_500.mmu.mapper26	PVBusMapper
MMU_500.mmu.mapper27	PVBusMapper
MMU_500.mmu.mapper28	PVBusMapper
MMU_500.mmu.mapper29	PVBusMapper
MMU_500.mmu.mapper3	PVBusMapper

InstanceName	ComponentName
MMU_500.mmu.mapper30	PVBusMapper
MMU_500.mmu.mapper31	PVBusMapper
MMU_500.mmu.mapper4	PVBusMapper
MMU_500.mmu.mapper5	PVBusMapper
MMU_500.mmu.mapper6	PVBusMapper
MMU_500.mmu.mapper7	PVBusMapper
MMU_500.mmu.mapper8	PVBusMapper
MMU_500.mmu.mapper9	PVBusMapper
MMU_500.mmu.ptw_dvm_receiver	PVBusMapper
MMU_500.mmu.ptw_master	PVBusMaster
MMU_500.mmu.pvbus_control_s_slv	PVBusSlave
MMU_500.mmu.pvbus_master	PVBusMaster
MMU_500.mmu.pvbusmaster0	PVBusMaster
MMU_500.mmu.pvbusmaster1	PVBusMaster
MMU_500.mmu.pvbusmaster10	PVBusMaster
MMU_500.mmu.pvbusmaster11	PVBusMaster
MMU_500.mmu.pvbusmaster12	PVBusMaster
MMU_500.mmu.pvbusmaster13	PVBusMaster
MMU_500.mmu.pvbusmaster14	PVBusMaster
MMU_500.mmu.pvbusmaster15	PVBusMaster
MMU_500.mmu.pvbusmaster16	PVBusMaster
MMU_500.mmu.pvbusmaster17	PVBusMaster
MMU_500.mmu.pvbusmaster18	PVBusMaster
MMU_500.mmu.pvbusmaster19	PVBusMaster
MMU_500.mmu.pvbusmaster2	PVBusMaster
MMU_500.mmu.pvbusmaster20	PVBusMaster
MMU_500.mmu.pvbusmaster21	PVBusMaster
MMU_500.mmu.pvbusmaster22	PVBusMaster
MMU_500.mmu.pvbusmaster23	PVBusMaster
MMU_500.mmu.pvbusmaster24	PVBusMaster
MMU_500.mmu.pvbusmaster25	PVBusMaster
MMU_500.mmu.pvbusmaster26	PVBusMaster
MMU_500.mmu.pvbusmaster27	PVBusMaster
MMU_500.mmu.pvbusmaster28	PVBusMaster
MMU_500.mmu.pvbusmaster29	PVBusMaster
MMU_500.mmu.pvbusmaster3	PVBusMaster
MMU_500.mmu.pvbusmaster30	PVBusMaster
MMU_500.mmu.pvbusmaster31	PVBusMaster
MMU_500.mmu.pvbusmaster4	PVBusMaster

InstanceName	ComponentName
MMU_500.mmu.pvbusmaster5	PVBusMaster
MMU_500.mmu.pvbusmaster6	PVBusMaster
MMU_500.mmu.pvbusmaster7	PVBusMaster
MMU_500.mmu.pvbusmaster8	PVBusMaster
MMU_500.mmu.pvbusmaster9	PVBusMaster
MMU_500.mmu.pvbusslave0	PVBusSlave
MMU_500.mmu.pvbusslave1	PVBusSlave
MMU_500.mmu.pvbusslave10	PVBusSlave
MMU_500.mmu.pvbusslave11	PVBusSlave
MMU_500.mmu.pvbusslave12	PVBusSlave
MMU_500.mmu.pvbusslave13	PVBusSlave
MMU_500.mmu.pvbusslave14	PVBusSlave
MMU_500.mmu.pvbusslave15	PVBusSlave
MMU_500.mmu.pvbusslave16	PVBusSlave
MMU_500.mmu.pvbusslave17	PVBusSlave
MMU_500.mmu.pvbusslave18	PVBusSlave
MMU_500.mmu.pvbusslave19	PVBusSlave
MMU_500.mmu.pvbusslave2	PVBusSlave
MMU_500.mmu.pvbusslave20	PVBusSlave
MMU_500.mmu.pvbusslave21	PVBusSlave
MMU_500.mmu.pvbusslave22	PVBusSlave
MMU_500.mmu.pvbusslave23	PVBusSlave
MMU_500.mmu.pvbusslave24	PVBusSlave
MMU_500.mmu.pvbusslave25	PVBusSlave
MMU_500.mmu.pvbusslave26	PVBusSlave
MMU_500.mmu.pvbusslave27	PVBusSlave
MMU_500.mmu.pvbusslave28	PVBusSlave
MMU_500.mmu.pvbusslave29	PVBusSlave
MMU_500.mmu.pvbusslave3	PVBusSlave
MMU_500.mmu.pvbusslave30	PVBusSlave
MMU_500.mmu.pvbusslave31	PVBusSlave
MMU_500.mmu.pvbusslave4	PVBusSlave
MMU_500.mmu.pvbusslave5	PVBusSlave
MMU_500.mmu.pvbusslave6	PVBusSlave
MMU_500.mmu.pvbusslave7	PVBusSlave
MMU_500.mmu.pvbusslave8	PVBusSlave
MMU_500.mmu.pvbusslave9	PVBusSlave

## Ports for MMU\_500

Table 3-1086: Ports

Name	Protocol	Type	Description
cfg_cttw_in	Signal	Slave	Enables coherent page table walks.
comb_irpt_ns	Signal	Master	Non-secure combined interrupt.
comb_irpt_s	Signal	Master	Secure combined interrupt.
cxt_irpt[128]	Signal	Master	Context interrupt.
glbl_flt_irpt_ns	Signal	Master	Global Non-secure fault interrupt.
glbl_flt_irpt_s	Signal	Master	Global Secure fault interrupt.
priv_internals	MMU_500_Internals	Slave	For internal use only, please do not use.
pvbus_control_s	PVBus	Slave	Provides memory-mapped read write access to the control registers of the module.
pvbus_m[32]	PVBus	Master	For all memory accesses. One for each Translation Buffer Unit (TBU).
pvbus_ptw_m	PVBus	Master	If ptw_has_separate_port is true, use for page table walks.
pvbus_s[32]	PVBus	Slave	For transactions from PVBus master/decoder. One for each TBU.
reset_in	Signal	Slave	Reset signal.

## Parameters for MMU\_500

**always\_secure\_ssd\_indices**

## Type

string

## Default value

""

## Description

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

**cfg\_cttw**

## Type

bool

## Default value

0x1

## Description

Perform coherent page table walks.

**dump\_unpredictability\_in\_user\_flags**

## Type

bool

## Default value

0x0



**Description**

Override the user flags to encode unpredictable information (validation only).

**label0\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label0: Read SDD or SSD\_Index.

**label0\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label0: Read Stream ID.

**label0\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label0: Write SDD or SSD\_Index.

**label0\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label0: Write Stream ID.

**label10\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label10: Read SDD or SSD\_Index.

**label10\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label10: Read Stream ID.

**label10\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label10: Write SDD or SSD\_Index.

**label10\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label10: Write Stream ID.

**label11\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label11: Read SDD or SSD\_Index.

**label11\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label11: Read Stream ID.

**label11\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label11: Write SDD or SSD\_Index.

**label11\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label11: Write Stream ID.

**label12\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label12: Read SDD or SSD\_Index.

**label12\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label12: Read Stream ID.

**label12\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label12: Write SDD or SSD\_Index.

**label12\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label12: Write Stream ID.

**label13\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label13: Read SDD or SSD\_Index.

**label13\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label13: Read Stream ID.

**label13\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label13: Write SDD or SSD\_Index.

**label13\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label13: Write Stream ID.

**label14\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label14: Read SDD or SSD\_Index.

**label14\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label14: Read Stream ID.

**label14\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label14: Write SDD or SSD\_Index.

**label14\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label14: Write Stream ID.

**label15\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label15: Read SDD or SSD\_Index.

**label15\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label15: Read Stream ID.

**label15\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label15: Write SDD or SSD\_Index.

**label15\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label15: Write Stream ID.

**label16\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label16: Read SDD or SSD\_Index.

**label16\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label16: Read Stream ID.

**label16\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label16: Write SDD or SSD\_Index.

**label16\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label16: Write Stream ID.

**label17\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label17: Read SDD or SSD\_Index.

**label17\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label17: Read Stream ID.

**label17\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label17: Write SDD or SSD\_Index.

**label17\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label17: Write Stream ID.

**label18\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label18: Read SDD or SSD\_Index.

**label18\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label18: Read Stream ID.

**label18\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label18: Write SDD or SSD\_Index.

**label18\_write\_stream\_id****Type**

int

**Default value**

0x0



**Description**

Label18: Write Stream ID.

**label19\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label19: Read SDD or SSD\_Index.

**label19\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label19: Read Stream ID.

**label19\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label19: Write SDD or SSD\_Index.

**label19\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label19: Write Stream ID.

**label1\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label1: Read SDD or SSD\_Index.

**label11\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label1: Read Stream ID.

**label11\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label1: Write SDD or SSD\_Index.

**label11\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label1: Write Stream ID.

**label120\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label20: Read SDD or SSD\_Index.

**label120\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label20: Read Stream ID.

**label20\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label20: Write SDD or SSD\_Index.

**label20\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label20: Write Stream ID.

**label21\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label21: Read SDD or SSD\_Index.

**label21\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label21: Read Stream ID.

**label21\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label21: Write SDD or SSD\_Index.

**label21\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label21: Write Stream ID.

**label22\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label22: Read SDD or SSD\_Index.

**label22\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label22: Read Stream ID.

**label22\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label22: Write SDD or SSD\_Index.

**label22\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label22: Write Stream ID.

**label123\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label23: Read SDD or SSD\_Index.

**label123\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label23: Read Stream ID.

**label123\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label23: Write SDD or SSD\_Index.

**label123\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label23: Write Stream ID.

**label124\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label24: Read SDD or SSD\_Index.

**label124\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label24: Read Stream ID.

**label124\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label24: Write SDD or SSD\_Index.

**label124\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label24: Write Stream ID.

**label125\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label25: Read SDD or SSD\_Index.

**label125\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label25: Read Stream ID.

**label25\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label25: Write SDD or SSD\_Index.

**label25\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label25: Write Stream ID.

**label26\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label26: Read SDD or SSD\_Index.

**label26\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label26: Read Stream ID.

**label26\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label26: Write SDD or SSD\_Index.

**label26\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label26: Write Stream ID.

**label27\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label27: Read SDD or SSD\_Index.

**label27\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label27: Read Stream ID.

**label27\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label27: Write SDD or SSD\_Index.

**label27\_write\_stream\_id****Type**

int

**Default value**

0x0



**Description**

Label27: Write Stream ID.

**label28\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label28: Read SDD or SSD\_Index.

**label28\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label28: Read Stream ID.

**label28\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label28: Write SDD or SSD\_Index.

**label28\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label28: Write Stream ID.

**label29\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label29: Read SDD or SSD\_Index.

**label129\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label29: Read Stream ID.

**label129\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label29: Write SDD or SSD\_Index.

**label129\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label29: Write Stream ID.

**label12\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label2: Read SDD or SSD\_Index.

**label12\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label2: Read Stream ID.

**label12\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label2: Write SDD or SSD\_Index.

**label12\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label2: Write Stream ID.

**label130\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label30: Read SDD or SSD\_Index.

**label130\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label30: Read Stream ID.

**label130\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label30: Write SDD or SSD\_Index.

**label30\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label30: Write Stream ID.

**label31\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label31: Read SDD or SSD\_Index.

**label31\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label31: Read Stream ID.

**label31\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label31: Write SDD or SSD\_Index.

**label31\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label31: Write Stream ID.

**label13\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label3: Read SDD or SSD\_Index.

**label13\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label3: Read Stream ID.

**label13\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label3: Write SDD or SSD\_Index.

**label13\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label3: Write Stream ID.

**label14\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label4: Read SDD or SSD\_Index.

**label14\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label4: Read Stream ID.

**label14\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label4: Write SDD or SSD\_Index.

**label14\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label4: Write Stream ID.

**label15\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label5: Read SDD or SSD\_Index.

**label15\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label5: Read Stream ID.

**label5\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label5: Write SDD or SSD\_Index.

**label5\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label5: Write Stream ID.

**label6\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label6: Read SDD or SSD\_Index.

**label6\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label6: Read Stream ID.

**label6\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label6: Write SDD or SSD\_Index.

**label6\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label6: Write Stream ID.

**label7\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label7: Read SDD or SSD\_Index.

**label7\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label7: Read Stream ID.

**label7\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label7: Write SDD or SSD\_Index.

**label7\_write\_stream\_id****Type**

int

**Default value**

0x0



**Description**

Label7: Write Stream ID.

**label18\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label8: Read SDD or SSD\_Index.

**label18\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label8: Read Stream ID.

**label18\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label8: Write SDD or SSD\_Index.

**label18\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label8: Write Stream ID.

**label19\_read\_ssd****Type**

int

**Default value**

0x0

**Description**

Label9: Read SDD or SSD\_Index.

**label9\_read\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label9: Read Stream ID.

**label9\_write\_ssd****Type**

int

**Default value**

0x0

**Description**

Label9: Write SDD or SSD\_Index.

**label9\_write\_stream\_id****Type**

int

**Default value**

0x0

**Description**

Label9: Write Stream ID.

**mmu.PRIVATE\_PARAMETER\_personality****Type**

string

**Default value**

""

**Description**

The personality to use (affects ID codes and various imp def features).

**mmu.PRIVATE\_PARAMETER\_seed****Type**

int

**Default value**

0x12345678

**Description**

Seed for randomised SMMU implementation defined behaviour.

**`mmu.PRIVATE_PARAMETER_validation_mode`****Type**

int

**Default value**

0x0

**Description**

Internal validation mode.

**`number_of_contexts`****Type**

int

**Default value**

0x8

**Description**

Number of context banks.

**`number_of_smrs`****Type**

int

**Default value**

0x20

**Description**

Number of stream match registers.

**`percent_tlbstatus_commits`****Type**

int

**Default value**

0xa

**Description**

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

**`prefetch_only_requests`****Type**

int

**Default value**

0x0

**Description**

Handle prefetch-only requests by:- 0 -- deny them 1 -- use debug table walks and TLB entries 2 -- treat them as normal transactions (dangerous).

**programmable\_non\_secure\_by\_default\_ssd\_indices****Type**

string

**Default value**

""

**Description**

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

**programmable\_secure\_by\_default\_ssd\_indices****Type**

string

**Default value**

""

**Description**

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

**ptw\_has\_separate\_port****Type**

bool

**Default value**

0x1

**Description**

Page Table Walks use pvbus\_ptw\_m.

**supports\_nested\_translations****Type**

bool

**Default value**

0x1

**Description**

Supports nested translations (stage 1 + stage 2).

**tlb\_depth****Type**

int

**Default value**  
0x800

**Description**  
TLB Depth (0 means 10000). The model will perform best with more TLB entries.

**use\_label\_mapping**

**Type**  
bool

**Default value**  
0x1

**Description**  
Use label mapping.

**use\_ssd\_determination\_table**

**Type**  
bool

**Default value**  
0x1

**Description**  
Use SSD Determination Table.

**version**

**Type**  
string

**Default value**  
"EAC"

**Description**  
Version of the RTL that the model represents. Valid values are LACr1 and EAC.

3.10.61 MMU\_500\_BASE

MMU-500 base component. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1087: IP revisions support

Revision	Quality level
r2p4	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Iris and MTI instances for MMU\_500\_BASE

This model has the following Iris instances:

**Table 3-1088: MMU\_500\_BASE Iris instances**

InstanceName	ComponentName
MMU_500_BASE	MMU_500_BASE
MMU_500_BASE.mapper0	PVBusMapper
MMU_500_BASE.mapper1	PVBusMapper
MMU_500_BASE.mapper10	PVBusMapper
MMU_500_BASE.mapper11	PVBusMapper
MMU_500_BASE.mapper12	PVBusMapper
MMU_500_BASE.mapper13	PVBusMapper
MMU_500_BASE.mapper14	PVBusMapper
MMU_500_BASE.mapper15	PVBusMapper
MMU_500_BASE.mapper16	PVBusMapper
MMU_500_BASE.mapper17	PVBusMapper
MMU_500_BASE.mapper18	PVBusMapper
MMU_500_BASE.mapper19	PVBusMapper
MMU_500_BASE.mapper2	PVBusMapper
MMU_500_BASE.mapper20	PVBusMapper
MMU_500_BASE.mapper21	PVBusMapper
MMU_500_BASE.mapper22	PVBusMapper
MMU_500_BASE.mapper23	PVBusMapper
MMU_500_BASE.mapper24	PVBusMapper
MMU_500_BASE.mapper25	PVBusMapper
MMU_500_BASE.mapper26	PVBusMapper
MMU_500_BASE.mapper27	PVBusMapper
MMU_500_BASE.mapper28	PVBusMapper
MMU_500_BASE.mapper29	PVBusMapper
MMU_500_BASE.mapper3	PVBusMapper
MMU_500_BASE.mapper30	PVBusMapper
MMU_500_BASE.mapper31	PVBusMapper
MMU_500_BASE.mapper4	PVBusMapper
MMU_500_BASE.mapper5	PVBusMapper
MMU_500_BASE.mapper6	PVBusMapper
MMU_500_BASE.mapper7	PVBusMapper
MMU_500_BASE.mapper8	PVBusMapper
MMU_500_BASE.mapper9	PVBusMapper
MMU_500_BASE.ptw_dvm_receiver	PVBusMapper
MMU_500_BASE.ptw_master	PVBusMaster

InstanceName	ComponentName
MMU_500_BASE.pvbus_control_s_slv	PVBusSlave
MMU_500_BASE.pvbus_master	PVBusMaster
MMU_500_BASE.pvbusmaster0	PVBusMaster
MMU_500_BASE.pvbusmaster1	PVBusMaster
MMU_500_BASE.pvbusmaster10	PVBusMaster
MMU_500_BASE.pvbusmaster11	PVBusMaster
MMU_500_BASE.pvbusmaster12	PVBusMaster
MMU_500_BASE.pvbusmaster13	PVBusMaster
MMU_500_BASE.pvbusmaster14	PVBusMaster
MMU_500_BASE.pvbusmaster15	PVBusMaster
MMU_500_BASE.pvbusmaster16	PVBusMaster
MMU_500_BASE.pvbusmaster17	PVBusMaster
MMU_500_BASE.pvbusmaster18	PVBusMaster
MMU_500_BASE.pvbusmaster19	PVBusMaster
MMU_500_BASE.pvbusmaster2	PVBusMaster
MMU_500_BASE.pvbusmaster20	PVBusMaster
MMU_500_BASE.pvbusmaster21	PVBusMaster
MMU_500_BASE.pvbusmaster22	PVBusMaster
MMU_500_BASE.pvbusmaster23	PVBusMaster
MMU_500_BASE.pvbusmaster24	PVBusMaster
MMU_500_BASE.pvbusmaster25	PVBusMaster
MMU_500_BASE.pvbusmaster26	PVBusMaster
MMU_500_BASE.pvbusmaster27	PVBusMaster
MMU_500_BASE.pvbusmaster28	PVBusMaster
MMU_500_BASE.pvbusmaster29	PVBusMaster
MMU_500_BASE.pvbusmaster3	PVBusMaster
MMU_500_BASE.pvbusmaster30	PVBusMaster
MMU_500_BASE.pvbusmaster31	PVBusMaster
MMU_500_BASE.pvbusmaster4	PVBusMaster
MMU_500_BASE.pvbusmaster5	PVBusMaster
MMU_500_BASE.pvbusmaster6	PVBusMaster
MMU_500_BASE.pvbusmaster7	PVBusMaster
MMU_500_BASE.pvbusmaster8	PVBusMaster
MMU_500_BASE.pvbusmaster9	PVBusMaster
MMU_500_BASE.pvbusslave0	PVBusSlave
MMU_500_BASE.pvbusslave1	PVBusSlave
MMU_500_BASE.pvbusslave10	PVBusSlave
MMU_500_BASE.pvbusslave11	PVBusSlave
MMU_500_BASE.pvbusslave12	PVBusSlave

InstanceName	ComponentName
MMU_500_BASE.pvbuslave13	PVBusSlave
MMU_500_BASE.pvbuslave14	PVBusSlave
MMU_500_BASE.pvbuslave15	PVBusSlave
MMU_500_BASE.pvbuslave16	PVBusSlave
MMU_500_BASE.pvbuslave17	PVBusSlave
MMU_500_BASE.pvbuslave18	PVBusSlave
MMU_500_BASE.pvbuslave19	PVBusSlave
MMU_500_BASE.pvbuslave2	PVBusSlave
MMU_500_BASE.pvbuslave20	PVBusSlave
MMU_500_BASE.pvbuslave21	PVBusSlave
MMU_500_BASE.pvbuslave22	PVBusSlave
MMU_500_BASE.pvbuslave23	PVBusSlave
MMU_500_BASE.pvbuslave24	PVBusSlave
MMU_500_BASE.pvbuslave25	PVBusSlave
MMU_500_BASE.pvbuslave26	PVBusSlave
MMU_500_BASE.pvbuslave27	PVBusSlave
MMU_500_BASE.pvbuslave28	PVBusSlave
MMU_500_BASE.pvbuslave29	PVBusSlave
MMU_500_BASE.pvbuslave3	PVBusSlave
MMU_500_BASE.pvbuslave30	PVBusSlave
MMU_500_BASE.pvbuslave31	PVBusSlave
MMU_500_BASE.pvbuslave4	PVBusSlave
MMU_500_BASE.pvbuslave5	PVBusSlave
MMU_500_BASE.pvbuslave6	PVBusSlave
MMU_500_BASE.pvbuslave7	PVBusSlave
MMU_500_BASE.pvbuslave8	PVBusSlave
MMU_500_BASE.pvbuslave9	PVBusSlave

This model has the following MTI trace components:

**Table 3-1089: MMU\_500\_BASE MTI instances**

InstanceName	ComponentName
MMU_500_BASE	MMU_500_BASE
MMU_500_BASE.mapper0	PVBusMapper
MMU_500_BASE.mapper1	PVBusMapper
MMU_500_BASE.mapper10	PVBusMapper
MMU_500_BASE.mapper11	PVBusMapper
MMU_500_BASE.mapper12	PVBusMapper
MMU_500_BASE.mapper13	PVBusMapper
MMU_500_BASE.mapper14	PVBusMapper



InstanceName	ComponentName
MMU_500_BASE.mapper15	PVBusMapper
MMU_500_BASE.mapper16	PVBusMapper
MMU_500_BASE.mapper17	PVBusMapper
MMU_500_BASE.mapper18	PVBusMapper
MMU_500_BASE.mapper19	PVBusMapper
MMU_500_BASE.mapper2	PVBusMapper
MMU_500_BASE.mapper20	PVBusMapper
MMU_500_BASE.mapper21	PVBusMapper
MMU_500_BASE.mapper22	PVBusMapper
MMU_500_BASE.mapper23	PVBusMapper
MMU_500_BASE.mapper24	PVBusMapper
MMU_500_BASE.mapper25	PVBusMapper
MMU_500_BASE.mapper26	PVBusMapper
MMU_500_BASE.mapper27	PVBusMapper
MMU_500_BASE.mapper28	PVBusMapper
MMU_500_BASE.mapper29	PVBusMapper
MMU_500_BASE.mapper3	PVBusMapper
MMU_500_BASE.mapper30	PVBusMapper
MMU_500_BASE.mapper31	PVBusMapper
MMU_500_BASE.mapper4	PVBusMapper
MMU_500_BASE.mapper5	PVBusMapper
MMU_500_BASE.mapper6	PVBusMapper
MMU_500_BASE.mapper7	PVBusMapper
MMU_500_BASE.mapper8	PVBusMapper
MMU_500_BASE.mapper9	PVBusMapper
MMU_500_BASE.ptw_dvm_receiver	PVBusMapper
MMU_500_BASE.ptw_master	PVBusMaster
MMU_500_BASE.pvbus_control_s_slv	PVBusSlave
MMU_500_BASE.pvbus_master	PVBusMaster
MMU_500_BASE.pvbusmaster0	PVBusMaster
MMU_500_BASE.pvbusmaster1	PVBusMaster
MMU_500_BASE.pvbusmaster10	PVBusMaster
MMU_500_BASE.pvbusmaster11	PVBusMaster
MMU_500_BASE.pvbusmaster12	PVBusMaster
MMU_500_BASE.pvbusmaster13	PVBusMaster
MMU_500_BASE.pvbusmaster14	PVBusMaster
MMU_500_BASE.pvbusmaster15	PVBusMaster
MMU_500_BASE.pvbusmaster16	PVBusMaster
MMU_500_BASE.pvbusmaster17	PVBusMaster

InstanceName	ComponentName
MMU_500_BASE.pvbusmaster18	PVBusMaster
MMU_500_BASE.pvbusmaster19	PVBusMaster
MMU_500_BASE.pvbusmaster2	PVBusMaster
MMU_500_BASE.pvbusmaster20	PVBusMaster
MMU_500_BASE.pvbusmaster21	PVBusMaster
MMU_500_BASE.pvbusmaster22	PVBusMaster
MMU_500_BASE.pvbusmaster23	PVBusMaster
MMU_500_BASE.pvbusmaster24	PVBusMaster
MMU_500_BASE.pvbusmaster25	PVBusMaster
MMU_500_BASE.pvbusmaster26	PVBusMaster
MMU_500_BASE.pvbusmaster27	PVBusMaster
MMU_500_BASE.pvbusmaster28	PVBusMaster
MMU_500_BASE.pvbusmaster29	PVBusMaster
MMU_500_BASE.pvbusmaster3	PVBusMaster
MMU_500_BASE.pvbusmaster30	PVBusMaster
MMU_500_BASE.pvbusmaster31	PVBusMaster
MMU_500_BASE.pvbusmaster4	PVBusMaster
MMU_500_BASE.pvbusmaster5	PVBusMaster
MMU_500_BASE.pvbusmaster6	PVBusMaster
MMU_500_BASE.pvbusmaster7	PVBusMaster
MMU_500_BASE.pvbusmaster8	PVBusMaster
MMU_500_BASE.pvbusmaster9	PVBusMaster
MMU_500_BASE.pvbusslave0	PVBusSlave
MMU_500_BASE.pvbusslave1	PVBusSlave
MMU_500_BASE.pvbusslave10	PVBusSlave
MMU_500_BASE.pvbusslave11	PVBusSlave
MMU_500_BASE.pvbusslave12	PVBusSlave
MMU_500_BASE.pvbusslave13	PVBusSlave
MMU_500_BASE.pvbusslave14	PVBusSlave
MMU_500_BASE.pvbusslave15	PVBusSlave
MMU_500_BASE.pvbusslave16	PVBusSlave
MMU_500_BASE.pvbusslave17	PVBusSlave
MMU_500_BASE.pvbusslave18	PVBusSlave
MMU_500_BASE.pvbusslave19	PVBusSlave
MMU_500_BASE.pvbusslave2	PVBusSlave
MMU_500_BASE.pvbusslave20	PVBusSlave
MMU_500_BASE.pvbusslave21	PVBusSlave
MMU_500_BASE.pvbusslave22	PVBusSlave
MMU_500_BASE.pvbusslave23	PVBusSlave

InstanceName	ComponentName
MMU_500_BASE.pvbuslave24	PVBusSlave
MMU_500_BASE.pvbuslave25	PVBusSlave
MMU_500_BASE.pvbuslave26	PVBusSlave
MMU_500_BASE.pvbuslave27	PVBusSlave
MMU_500_BASE.pvbuslave28	PVBusSlave
MMU_500_BASE.pvbuslave29	PVBusSlave
MMU_500_BASE.pvbuslave3	PVBusSlave
MMU_500_BASE.pvbuslave30	PVBusSlave
MMU_500_BASE.pvbuslave31	PVBusSlave
MMU_500_BASE.pvbuslave4	PVBusSlave
MMU_500_BASE.pvbuslave5	PVBusSlave
MMU_500_BASE.pvbuslave6	PVBusSlave
MMU_500_BASE.pvbuslave7	PVBusSlave
MMU_500_BASE.pvbuslave8	PVBusSlave
MMU_500_BASE.pvbuslave9	PVBusSlave

## Ports for MMU\_500\_BASE

**Table 3-1090: Ports**

Name	Protocol	Type	Description
cfg_cttw_in	Signal	Slave	The SoC supports coherent page walks, this is meant to be sampled at reset. However, in practice the model has to prevent the race condition between cfg_cttw being asserted at the same 'cycle' as negedge reset. Thus we actually only sample the signal on the first transaction to the SMMU or the first transition on this signal after reset. Thus in the model, we require that cfg_cttw be held for at least this period of time.
comb_irpt_ns	Signal	Master	"Non-secure combined interrupt"
comb_irpt_s	Signal	Master	"Secure combined interrupt"
cxt_irpt[128]	Signal	Master	Non-secure context bank fault.
glblflt_irpt_ns	Signal	Master	Global non-secure Fault Interrupt In the SMMU Architecture this is called SMMU_NSglrpt.
glblflt_irpt_s	Signal	Master	Global secure Fault Interrupt In the SMMU Architecture this is called SMMU_glrpt.
identify	MMU_500_BASE_IDENTIFY	Master	This port is a special model port that is used to take a transaction and map it to an SSD/SSD_Index and StreamID.
priv_internals	MMU_500_Internals	Slave	For internal use only, please do not use.
pvbus_control_s	PVBus	Slave	The register port of the device is AXI.
pvbus_m[32]	PVBus	Master	This downstream port is where the translated accesses from pvbus_s emerge. See notes for pvbus_s[] as well. If the Page Table Walk (PTW) does not have a separate port then PTW accesses will emerge at port 0 with the same attributes as described in pvbus_ptw_m.

Name	Protocol	Type	Description
pvbus_ptw_m	PVBus	Master	This downstream port is where page table walk accesses come from. This is only used if configured to use a separate page table walk port. The MMU-500 will only obey DVM messages if configured to use this port. The page walks come out of this port with the following master_id and user_flags. master_id : 0xFFFFFFFF The user flags : user_flags[7:0] stage 1 context_id (or 0xFF if stage2 only) user_flags[15:8] stage 2 context_id (or 0xFF if stage 1 with stage 2 bypass) user_flags[18:16] stage 1 level user_flags[21:19] stage 2 level user_flags[31,30] adomain of the transaction NOTE that if the walk is being done for a stage 1 page walk descriptor fetch then the stage 1 level field will indicate that level. If the walk is being done for a stage 2 descriptor fetch, then the stage 2 level field will show that level. If the context-id for a stage is not valid (0xFF) then the 'level' information is 0x7.
pvbus_s[32]	PVBus	Slave	This port is the upstream port of the device, addresses on the port are in the VA/IPA Each TBU in the design is represented by a pair of pvbus_s[tbu_id] and pvbus_m[tbu_id]. That is transactions that go into pvbus_s[tbu_id] will emerge at pvbus_m[tbu_id]. The port index that a transaction comes in on is the tbu_number_ parameter to the MMU_500_BASE_IDENTIFY::identify() function. The identify() function must use all the information it is given by the parameters to map to the architectural concepts of StreamID and SSD_Index/SSD. How it does this is IMPLEMENTATION DEFINED and depends on the topology of the SoC and the masters upstream of the TBUs.
reset_in	Signal	Slave	The reset pin.

## Parameters for MMU\_500\_BASE

### PRIVATE\_PARAMETER\_personality

#### Type

string

#### Default value

""

#### Description

The personality to use (affects ID codes and various imp def features).

### PRIVATE\_PARAMETER\_seed

#### Type

int

#### Default value

0x12345678

#### Description

Seed for randomised SMMU implementation defined behaviour.

### PRIVATE\_PARAMETER\_validation\_mode

#### Type

int

**Default value**

0x0

**Description**

Internal validation mode.

**`always_secure_ssd_indices`****Type**

string

**Default value**

""

**Description**

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

**`cfg_cttw`****Type**

bool

**Default value**

0x1

**Description**

Perform coherent page table walks.

**`dump_unpredictability_in_user_flags`****Type**

bool

**Default value**

0x0

**Description**

Override the user flags to encode unpredictable information (validation only).

**`number_of_contexts`****Type**

int

**Default value**

0x8

**Description**

Number of context banks.

**`number_of_smrs`****Type**

int

**Default value**

0x10

**Description**

Number of stream match registers.

**percent\_tlbstatus\_commits****Type**

int

**Default value**

0xa

**Description**

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

**prefetch\_only\_requests****Type**

int

**Default value**

0x0

**Description**

Handle prefetch-only requests by:- 0 -- deny them 1 -- use debug table walks and TLB entries 2 -- treat them as normal transactions (dangerous).

**programmable\_non\_secure\_by\_default\_ssd\_indices****Type**

string

**Default value**

""

**Description**

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

**programmable\_secure\_by\_default\_ssd\_indices****Type**

string

**Default value**

""

**Description**

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

**ptw\_has\_separate\_port****Type**

bool

**Default value**

0x1

**Description**

Page Table Walks use pvbus\_ptw\_m (or uses pvbus\_m[0]).

**supports\_nested\_translations****Type**

bool

**Default value**

0x1

**Description**

Supports nested translations (stage 1 + stage 2).

**tlb\_depth****Type**

int

**Default value**

0x800

**Description**

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

**use\_ssd\_determination\_table****Type**

bool

**Default value**

0x1

**Description**

Use SSD Determination Table.

**version****Type**

string

**Default value**

"EAC"

**Description**

Version of the RTL that the model represents. Valid values are LACr1 and EAC.

### 3.10.62 MMU\_600

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1091: IP revisions support**

Revision	Quality level
r0p0	Full support
r0p1	Full support
r0p2	Full support
r1p0	Full support
r2p0	Full support
r2p1	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Limitations

- No power control
- AMBA stash operations, destructive read and destructive hint operations are not supported on PVBus and also are not supported by the device.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMC_G_CEIFD0` fields. The PMU is intended for demonstration purposes only and for driver development.



Note

Between 11.17 and 11.18, the point of triggering for events changed for 'TLB miss' and this might lead to an (architecturally valid) change in the values captured in some circumstances.

- Limited RAS support
- The SYSCO interface is not implemented
- The low power interface is not implemented
- `TCU_CFG.XLATE_SLOTS` is fixed at 512
- `TCU_STATUS.GNT_XLATE_SLOTS` always reads at 512

#### Notes

- A bad configuration renders the model inactive.
- Some configurations can be adjusted by configuration pins. These are only sampled at the negative edge of the reset pin. If you want to use these pins then you must drive them before sending a negative edge on the reset pin. During simulation reset the component driving them must also drive this transition again.



- Debug reads to the registers do not disturb the state.
- Writes to registers with Update flags, including debug writes, are ignored if the Update flag is already set to one.
- Debug and real accesses to the registers must be 32 bits or 64 bits.
- The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping used is remembered by the bus infrastructure and is used for subsequent sufficiently similar transactions without requiring intervention from the SMMU model, and is not traced, for instance.

The range of addresses that the mapping is valid for is determined by the SMMU model, depending on architectural and model-implementation details. However, as it is unaware of any sign-extension or the mapping that the SoC does, the SoC is responsible for subsequently narrowing the range of addresses for which this mapping is valid. Typically, this is done automatically when using PVBUSMapper.

- The hardware is a distributed SMMU and is divided into:
  - A single Translation Control Unit (TCU)
    - Has a port for the programming interface of the SMMU
    - Receives DVM messages
    - Does all the page walking, queue manipulation, etc.
  - One or more Translation Bus Units (TBUs)
    - Translate transactions from upstream (client) device into downstream transactions.
  - Zero or more connections to PCIe Root Complexes (PCIe-RCs)
  - There can be a total of 62 TBUs and PCIe-RCs attached.
  - An interconnect connecting the TBUs, PCIe-RCs to the TCU.
- A PCIe-RC has one connection to the TCU to make ATS requests but the PCIe-RC uses one or more TBUs to transform the transactions and pass them to the memory system. In the model, those TBUs are configured by the parameter `list_of_pcie_mode`. The SMMU does not know which TBUs a particular PCIe-RC is attached to.
- The TBU ORs a value into each StreamID that it receives. In the model, this is configured by the parameters:
  - `list_of_s_sid_high_at_bitpos0`
  - `list_of_ns_sid_high_at_bitpos0`
- The TCU, TBU, and the interconnect are all represented by this single model component.
- In the model, a pair of ports `tbs_pvbuss_s[i]/tbm_pvbuss_m[i]` represent a TBU 'i' or the `tbs_pvbuss_s[i]` represents an incoming connection for a PCIe-RC. The corresponding reverse connection from the TCU to the PCIe-RC is by a special bus called `pvbus_id_routed_m` that is used to transport ATC Invalidates to the PCIe-RC.
  - In order to reduce system construction complexity the `tbs_pvbuss_s[i]/tbm_pvbuss_m[i]` also acts as a TBU so that the PCIe-RC need not separate its normal transactions and its ATS requests.

- However, ATC Invalidates are only sent to a port which appears in `list_of_pcie_rc`. It should be uniquely decoded to a single port based on `list_of_ns_sid_high_at_bitpos0` and those ATC Invalidates must be routed to the correct PCIe subsystem in order to invalidate the cache of ATS Response in the subsystem. Thus all TBUs that a PCIe-RC uses must have a unique reverse mapping from stream id to port.
- The pin `sup_oas` is not supported, instead it is a parameter as it is assumed that it would be tied to a fixed value in any specific platform.
- The hardware only has a single cacheability attribute for input transactions, but PVBUS transports both inner and outer cacheability.
- For non-PCIe-mode TBUs (i.e. their index does not appear in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - For non-cache maintenance operations:
    - If the input attribute is any type of device then it is well defined as being outer-shared and Device-nGnRnE or Device-nGnRE. There is no support for Gathering or Reordering.
    - If the outer cacheable input attribute is normal then if it is Write-back then this is converted to Inner Write-back Outer Write-back (iWB-oWB) with the desired shareability. No Transient hint is supported and is always treated as non-transient.
    - This leaves all other normal memory types that are mapped to Inner Normal Non-cacheable, Outer Normal Non-cacheable outer shared (iNC-oNC-osh).
- Thus the upstream devices must present the cacheability in the *outer* cacheability attribute on PVBUS if it is cacheable. If it is a device type then both the inner and outer attributes must be set to the same. If it is iNC-oNC-osh then it must be presented as such.
- For PCIe-mode TBUs (i.e. their index appears in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - Input transactions are from PCIe and the only indication of the memory type is in the NoSnoop bit of the transaction. No shareability is transported.
    - NoSnoop interpreted as iNC-oNC-osh
    - ! NoSnoop interpreted as iWB-oWB-ish (note inner shareable)
  - If a NoSnoop transaction has an attribute transform applied to it and the result of the transform is weaker than iNC-oNC-osh then it is forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it is forced to iNC-oNC-osh. However, if the page tables transformed it to a device type then, as all device types are stronger than iNC-oNC-osh then it exits the SMMU as the device type.
  - In the model, transactions are classified by their incoming memory attributes as to whether they are NoSnoop or not and then normalized appropriately:
    - iWB-oWB-any-shareability transactions are interpreted as ! NoSnoop and therefore are normalized to iWB-oWB-ish.
    - Anything else is considered NoSnoop and therefore is normalized to iNC-oNC-osh.
  - Translated accesses also suffer the same interpretation to determine NoSnoop and how they are normalized. Thus they could enter the system with attributes different to if they were Untranslated Accesses translated by normal means.

- It is expected that there is a component downstream of the SMMU that is aware of the system address map and will override the memory type to 'device' for any transaction that accesses a peripheral.
- The hardware has a single cacheability on input and, for transactions that are neither cache-maintenance operations nor PCIe transactions, normalizes the input to an architectural form before performing the SMMUv3 architectural transform:
  - Any device type is left untouched (the input can only represent Device-nGnRE and Device-nGnRnE).
  - If the input is Write-back (WB) then it is normalized to iWB-oWB with the incoming shareability.
  - If the input is anything else it is normalized to iNC-oNC-osh.
- The model accepts full architectural attributes of two levels of cacheability and so has to decide how to interpret this in terms of the hardware. For transactions that are not cache maintenance operations, the model replicates the outer attribute into the inner attribute and then performs the normalization that the hardware does.
- The hardware normalizes the architectural output attributes and outputs a single level of cacheability and a user flag (OC) specifying if the architectural attributes were cacheable in the outer cacheable domain. If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. If the original transaction was NoSnoop, then any weaker memory type is strengthened to iNC-oNC-osh so apply the following transform:

```

if      iWB-oWB-nsh/ish/osh      then output WB-nsh/ish/osh, OC = 1
else if i (NC/WT/WB) -o (WB/WT)  then output NC-Sys,      OC = 1
else if i (NC/WT/WB) -oNC        then output NC-Sys,      OC = 0
else if Device-(GRE/nGRE/nGnRE) then output DV-Sys,      OC = 0
else                               output SO-Sys,      OC = 0

```

- The model only normalizes according to PCIe but otherwise leaves the architectural attributes intact on the output bus.
- MSIs are issued on the `qtw_pvbus_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `MasterID=0xFFFFFFf`.

In the hardware, there is no way of distinguishing Event Queue writes from MSI writes, however, this provides a mechanism that if the model system needs to distinguish then it can.

## Parameters for parsing transaction attributes

Some of the parameters are strings that share a common parsing format for extracting from and placing information into the transaction's attributes.

They can extract and place information into the following fields of a transaction's attributes:

- `ManagerID64` (64 bits)
- `MasterID` (32 bits) (This is the lower 32 bits of `ManagerID64`)
- `ExtendedID` (64 bits)
- `UserFlags` (32 bits)

The string parameter is parsed as a comma-separated list of:

```
lhs_expression=rhs_expression
```

The `lhs_expression` and `rhs_expression` can be an entire symbol, for example:

```
StreamID
```

or a bit, or bit slice:

```
StreamID[16]  
StreamID[31:20]
```

In `rhs_expression`, numeric constants (or slices of numeric constants) are also allowed:

```
StreamID[16]=1  
StreamID[16]=10[2]
```

A single symbol might be assigned from multiple non-contiguous arrays of bits from a mix of different RHS symbols:

```
StreamID[16]=1, StreamID[31:30]=ExtendedID[1:0],  
StreamID[15:0]=UserFlags[31:16], ...
```

In those cases where a left hand symbol can also appear on the right hand side, it is possible to swap bits and transform the symbol. For example, the following expression swaps bits 0 and 1 of the `ExtendedID`:

```
ExtendedID[0]=ExtendedID[1], ExtendedID[1]=ExtendedID[0]
```

Any bits of the attributes that have no transform specified are retained from the input.

The `lhs_expression` and `rhs_expression` must have the same bit width.

## Iris and MTI instances for MMU\_600

This model has the following Iris instances:

**Table 3-1092: MMU\_600 Iris instances**

InstanceName	ComponentName
MMU_600	MMU_600
MMU_600.register_file[0]	PVBusSlave
MMU_600.service_request_tbu[0]	PVBusSlave
MMU_600.service_request_tbu[10]	PVBusSlave
MMU_600.service_request_tbu[11]	PVBusSlave
MMU_600.service_request_tbu[12]	PVBusSlave

InstanceName	ComponentName
MMU_600.service_request_tbu[13]	PVBusSlave
MMU_600.service_request_tbu[14]	PVBusSlave
MMU_600.service_request_tbu[15]	PVBusSlave
MMU_600.service_request_tbu[16]	PVBusSlave
MMU_600.service_request_tbu[17]	PVBusSlave
MMU_600.service_request_tbu[18]	PVBusSlave
MMU_600.service_request_tbu[19]	PVBusSlave
MMU_600.service_request_tbu[1]	PVBusSlave
MMU_600.service_request_tbu[20]	PVBusSlave
MMU_600.service_request_tbu[21]	PVBusSlave
MMU_600.service_request_tbu[22]	PVBusSlave
MMU_600.service_request_tbu[23]	PVBusSlave
MMU_600.service_request_tbu[24]	PVBusSlave
MMU_600.service_request_tbu[25]	PVBusSlave
MMU_600.service_request_tbu[26]	PVBusSlave
MMU_600.service_request_tbu[27]	PVBusSlave
MMU_600.service_request_tbu[28]	PVBusSlave
MMU_600.service_request_tbu[29]	PVBusSlave
MMU_600.service_request_tbu[2]	PVBusSlave
MMU_600.service_request_tbu[30]	PVBusSlave
MMU_600.service_request_tbu[31]	PVBusSlave
MMU_600.service_request_tbu[32]	PVBusSlave
MMU_600.service_request_tbu[33]	PVBusSlave
MMU_600.service_request_tbu[34]	PVBusSlave
MMU_600.service_request_tbu[35]	PVBusSlave
MMU_600.service_request_tbu[36]	PVBusSlave
MMU_600.service_request_tbu[37]	PVBusSlave
MMU_600.service_request_tbu[38]	PVBusSlave
MMU_600.service_request_tbu[39]	PVBusSlave
MMU_600.service_request_tbu[3]	PVBusSlave
MMU_600.service_request_tbu[40]	PVBusSlave
MMU_600.service_request_tbu[41]	PVBusSlave
MMU_600.service_request_tbu[42]	PVBusSlave
MMU_600.service_request_tbu[43]	PVBusSlave
MMU_600.service_request_tbu[44]	PVBusSlave
MMU_600.service_request_tbu[45]	PVBusSlave
MMU_600.service_request_tbu[46]	PVBusSlave
MMU_600.service_request_tbu[47]	PVBusSlave
MMU_600.service_request_tbu[48]	PVBusSlave

InstanceName	ComponentName
MMU_600.service_request_tbu[49]	PVBusSlave
MMU_600.service_request_tbu[4]	PVBusSlave
MMU_600.service_request_tbu[50]	PVBusSlave
MMU_600.service_request_tbu[51]	PVBusSlave
MMU_600.service_request_tbu[52]	PVBusSlave
MMU_600.service_request_tbu[53]	PVBusSlave
MMU_600.service_request_tbu[54]	PVBusSlave
MMU_600.service_request_tbu[55]	PVBusSlave
MMU_600.service_request_tbu[56]	PVBusSlave
MMU_600.service_request_tbu[57]	PVBusSlave
MMU_600.service_request_tbu[58]	PVBusSlave
MMU_600.service_request_tbu[59]	PVBusSlave
MMU_600.service_request_tbu[5]	PVBusSlave
MMU_600.service_request_tbu[60]	PVBusSlave
MMU_600.service_request_tbu[61]	PVBusSlave
MMU_600.service_request_tbu[62]	PVBusSlave
MMU_600.service_request_tbu[63]	PVBusSlave
MMU_600.service_request_tbu[6]	PVBusSlave
MMU_600.service_request_tbu[7]	PVBusSlave
MMU_600.service_request_tbu[8]	PVBusSlave
MMU_600.service_request_tbu[9]	PVBusSlave
MMU_600.tbu[0]	PVBusMapper
MMU_600.tbu[10]	PVBusMapper
MMU_600.tbu[11]	PVBusMapper
MMU_600.tbu[12]	PVBusMapper
MMU_600.tbu[13]	PVBusMapper
MMU_600.tbu[14]	PVBusMapper
MMU_600.tbu[15]	PVBusMapper
MMU_600.tbu[16]	PVBusMapper
MMU_600.tbu[17]	PVBusMapper
MMU_600.tbu[18]	PVBusMapper
MMU_600.tbu[19]	PVBusMapper
MMU_600.tbu[1]	PVBusMapper
MMU_600.tbu[20]	PVBusMapper
MMU_600.tbu[21]	PVBusMapper
MMU_600.tbu[22]	PVBusMapper
MMU_600.tbu[23]	PVBusMapper
MMU_600.tbu[24]	PVBusMapper
MMU_600.tbu[25]	PVBusMapper

InstanceName	ComponentName
MMU_600.tbu[26]	PVBusMapper
MMU_600.tbu[27]	PVBusMapper
MMU_600.tbu[28]	PVBusMapper
MMU_600.tbu[29]	PVBusMapper
MMU_600.tbu[2]	PVBusMapper
MMU_600.tbu[30]	PVBusMapper
MMU_600.tbu[31]	PVBusMapper
MMU_600.tbu[32]	PVBusMapper
MMU_600.tbu[33]	PVBusMapper
MMU_600.tbu[34]	PVBusMapper
MMU_600.tbu[35]	PVBusMapper
MMU_600.tbu[36]	PVBusMapper
MMU_600.tbu[37]	PVBusMapper
MMU_600.tbu[38]	PVBusMapper
MMU_600.tbu[39]	PVBusMapper
MMU_600.tbu[3]	PVBusMapper
MMU_600.tbu[40]	PVBusMapper
MMU_600.tbu[41]	PVBusMapper
MMU_600.tbu[42]	PVBusMapper
MMU_600.tbu[43]	PVBusMapper
MMU_600.tbu[44]	PVBusMapper
MMU_600.tbu[45]	PVBusMapper
MMU_600.tbu[46]	PVBusMapper
MMU_600.tbu[47]	PVBusMapper
MMU_600.tbu[48]	PVBusMapper
MMU_600.tbu[49]	PVBusMapper
MMU_600.tbu[4]	PVBusMapper
MMU_600.tbu[50]	PVBusMapper
MMU_600.tbu[51]	PVBusMapper
MMU_600.tbu[52]	PVBusMapper
MMU_600.tbu[53]	PVBusMapper
MMU_600.tbu[54]	PVBusMapper
MMU_600.tbu[55]	PVBusMapper
MMU_600.tbu[56]	PVBusMapper
MMU_600.tbu[57]	PVBusMapper
MMU_600.tbu[58]	PVBusMapper
MMU_600.tbu[59]	PVBusMapper
MMU_600.tbu[5]	PVBusMapper
MMU_600.tbu[60]	PVBusMapper

InstanceName	ComponentName
MMU_600.tbv[61]	PVBusMapper
MMU_600.tbv[62]	PVBusMapper
MMU_600.tbv[63]	PVBusMapper
MMU_600.tbv[6]	PVBusMapper
MMU_600.tbv[7]	PVBusMapper
MMU_600.tbv[8]	PVBusMapper
MMU_600.tbv[9]	PVBusMapper

This model has the following MTI trace components:

**Table 3-1093: MMU\_600 MTI instances**

InstanceName	ComponentName
MMU_600	MMU_600
MMU_600.register_file[0]	PVBusSlave
MMU_600.service_request_tbu[0]	PVBusSlave
MMU_600.service_request_tbu[10]	PVBusSlave
MMU_600.service_request_tbu[11]	PVBusSlave
MMU_600.service_request_tbu[12]	PVBusSlave
MMU_600.service_request_tbu[13]	PVBusSlave
MMU_600.service_request_tbu[14]	PVBusSlave
MMU_600.service_request_tbu[15]	PVBusSlave
MMU_600.service_request_tbu[16]	PVBusSlave
MMU_600.service_request_tbu[17]	PVBusSlave
MMU_600.service_request_tbu[18]	PVBusSlave
MMU_600.service_request_tbu[19]	PVBusSlave
MMU_600.service_request_tbu[1]	PVBusSlave
MMU_600.service_request_tbu[20]	PVBusSlave
MMU_600.service_request_tbu[21]	PVBusSlave
MMU_600.service_request_tbu[22]	PVBusSlave
MMU_600.service_request_tbu[23]	PVBusSlave
MMU_600.service_request_tbu[24]	PVBusSlave
MMU_600.service_request_tbu[25]	PVBusSlave
MMU_600.service_request_tbu[26]	PVBusSlave
MMU_600.service_request_tbu[27]	PVBusSlave
MMU_600.service_request_tbu[28]	PVBusSlave
MMU_600.service_request_tbu[29]	PVBusSlave
MMU_600.service_request_tbu[2]	PVBusSlave
MMU_600.service_request_tbu[30]	PVBusSlave
MMU_600.service_request_tbu[31]	PVBusSlave
MMU_600.service_request_tbu[32]	PVBusSlave



InstanceName	ComponentName
MMU_600.service_request_tbu[33]	PVBusSlave
MMU_600.service_request_tbu[34]	PVBusSlave
MMU_600.service_request_tbu[35]	PVBusSlave
MMU_600.service_request_tbu[36]	PVBusSlave
MMU_600.service_request_tbu[37]	PVBusSlave
MMU_600.service_request_tbu[38]	PVBusSlave
MMU_600.service_request_tbu[39]	PVBusSlave
MMU_600.service_request_tbu[3]	PVBusSlave
MMU_600.service_request_tbu[40]	PVBusSlave
MMU_600.service_request_tbu[41]	PVBusSlave
MMU_600.service_request_tbu[42]	PVBusSlave
MMU_600.service_request_tbu[43]	PVBusSlave
MMU_600.service_request_tbu[44]	PVBusSlave
MMU_600.service_request_tbu[45]	PVBusSlave
MMU_600.service_request_tbu[46]	PVBusSlave
MMU_600.service_request_tbu[47]	PVBusSlave
MMU_600.service_request_tbu[48]	PVBusSlave
MMU_600.service_request_tbu[49]	PVBusSlave
MMU_600.service_request_tbu[4]	PVBusSlave
MMU_600.service_request_tbu[50]	PVBusSlave
MMU_600.service_request_tbu[51]	PVBusSlave
MMU_600.service_request_tbu[52]	PVBusSlave
MMU_600.service_request_tbu[53]	PVBusSlave
MMU_600.service_request_tbu[54]	PVBusSlave
MMU_600.service_request_tbu[55]	PVBusSlave
MMU_600.service_request_tbu[56]	PVBusSlave
MMU_600.service_request_tbu[57]	PVBusSlave
MMU_600.service_request_tbu[58]	PVBusSlave
MMU_600.service_request_tbu[59]	PVBusSlave
MMU_600.service_request_tbu[5]	PVBusSlave
MMU_600.service_request_tbu[60]	PVBusSlave
MMU_600.service_request_tbu[61]	PVBusSlave
MMU_600.service_request_tbu[62]	PVBusSlave
MMU_600.service_request_tbu[63]	PVBusSlave
MMU_600.service_request_tbu[6]	PVBusSlave
MMU_600.service_request_tbu[7]	PVBusSlave
MMU_600.service_request_tbu[8]	PVBusSlave
MMU_600.service_request_tbu[9]	PVBusSlave
MMU_600.tbu[0]	PVBusMapper

InstanceName	ComponentName
MMU_600.tbu[10]	PVBusMapper
MMU_600.tbu[11]	PVBusMapper
MMU_600.tbu[12]	PVBusMapper
MMU_600.tbu[13]	PVBusMapper
MMU_600.tbu[14]	PVBusMapper
MMU_600.tbu[15]	PVBusMapper
MMU_600.tbu[16]	PVBusMapper
MMU_600.tbu[17]	PVBusMapper
MMU_600.tbu[18]	PVBusMapper
MMU_600.tbu[19]	PVBusMapper
MMU_600.tbu[1]	PVBusMapper
MMU_600.tbu[20]	PVBusMapper
MMU_600.tbu[21]	PVBusMapper
MMU_600.tbu[22]	PVBusMapper
MMU_600.tbu[23]	PVBusMapper
MMU_600.tbu[24]	PVBusMapper
MMU_600.tbu[25]	PVBusMapper
MMU_600.tbu[26]	PVBusMapper
MMU_600.tbu[27]	PVBusMapper
MMU_600.tbu[28]	PVBusMapper
MMU_600.tbu[29]	PVBusMapper
MMU_600.tbu[2]	PVBusMapper
MMU_600.tbu[30]	PVBusMapper
MMU_600.tbu[31]	PVBusMapper
MMU_600.tbu[32]	PVBusMapper
MMU_600.tbu[33]	PVBusMapper
MMU_600.tbu[34]	PVBusMapper
MMU_600.tbu[35]	PVBusMapper
MMU_600.tbu[36]	PVBusMapper
MMU_600.tbu[37]	PVBusMapper
MMU_600.tbu[38]	PVBusMapper
MMU_600.tbu[39]	PVBusMapper
MMU_600.tbu[3]	PVBusMapper
MMU_600.tbu[40]	PVBusMapper
MMU_600.tbu[41]	PVBusMapper
MMU_600.tbu[42]	PVBusMapper
MMU_600.tbu[43]	PVBusMapper
MMU_600.tbu[44]	PVBusMapper
MMU_600.tbu[45]	PVBusMapper

InstanceName	ComponentName
MMU_600.tbv[46]	PVBusMapper
MMU_600.tbv[47]	PVBusMapper
MMU_600.tbv[48]	PVBusMapper
MMU_600.tbv[49]	PVBusMapper
MMU_600.tbv[4]	PVBusMapper
MMU_600.tbv[50]	PVBusMapper
MMU_600.tbv[51]	PVBusMapper
MMU_600.tbv[52]	PVBusMapper
MMU_600.tbv[53]	PVBusMapper
MMU_600.tbv[54]	PVBusMapper
MMU_600.tbv[55]	PVBusMapper
MMU_600.tbv[56]	PVBusMapper
MMU_600.tbv[57]	PVBusMapper
MMU_600.tbv[58]	PVBusMapper
MMU_600.tbv[59]	PVBusMapper
MMU_600.tbv[5]	PVBusMapper
MMU_600.tbv[60]	PVBusMapper
MMU_600.tbv[61]	PVBusMapper
MMU_600.tbv[62]	PVBusMapper
MMU_600.tbv[63]	PVBusMapper
MMU_600.tbv[6]	PVBusMapper
MMU_600.tbv[7]	PVBusMapper
MMU_600.tbv[8]	PVBusMapper
MMU_600.tbv[9]	PVBusMapper

## Ports for MMU\_600

**Table 3-1094: Ports**

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock signal (in RTL aclk) This is a clock time-base used by the TCU to spread some of its processing over time, if enabled by the wait_* parameters. The clock must always be connected.
cmd_sync_irpt_ns	Signal	Master	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of SIG_IRQ.
cmd_sync_irpt_s	Signal	Master	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
event_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
event_q_irpt_s	Signal	Master	Pulsed interrupt output signal for the secure event queue becoming non-empty.
evento	Signal	Master	Event signal

Name	Protocol	Type	Description
global_irpt_ns	Signal	Master	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signalling an error.
global_irpt_s	Signal	Master	Pulsed interrupt output signal for secure SMMU_S_GERROR(N) signalling an error.
identify	SMMUv3AEMIdentifyProtocol	Master	Map the transaction to the tuple (StreamID, SubStreamID, SubStreamIDValid, SSD) The StreamID that is produced by the implementation of this protocol is not the final StreamID. The final StreamID is produced by using the list_of_ns_sid_high_at_bitpos0/list_of_s_sid_high_at_bitpos0 parameter to map the StreamID based on the upstream port index. Also see the parameter howto_identify which can replace the functionality of this port under certain circumstances.
pri_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the PRI queue becoming non-empty. Exists only for r1 and higher.
prog_pvbus_s	PVBus	Slave	Register subordinate port (in RTL PROG)
pvbus_id_routed_m[62]	PVBus	Master	This is a special "id-routed" port for transmitting ATC invalidates upstream into the PCIe EndPoints, it is not a normal bus. See the parameter output_id_routed_transform. The FastModels ATC Invalidate and PRI Response protocol specifies how to route and deal with this port.
qtw_pvbus_m	PVBus	Master	Manager port used for Table Walks, MSIs and Queue access. Note that although this is a manager port, it can still receive DVM messages.
sec_override	Signal	Slave	Allow certain registers to be accessible to non-secure accesses from reset, as described in the TCU_SCR register. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_btm	Signal	Slave	System supports BTM and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If BTM (Broadcast Table Maintenance) is not supported then DVM messages will be ignored. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_cohacc	Signal	Slave	System supports COHACC and will be reflected in the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_sev	Signal	Slave	System supports SEV and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
tbm_pvbus_m[62]	PVBus	Master	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered tbs_pvbus_s[] port.
tbs_pvbus_s[62]	PVBus	Slave	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
tbu_pmu_irpt[62]	Signal	Master	TBU Performance Monitoring Unit interrupt, one per TBU.
tbu_pmusnapshot_ack[62]	Signal	Master	PMU snapshot interface for the TBU, ack a snapshot.

Name	Protocol	Type	Description
tbu_pmusnapshot_req[62]	Signal	Slave	PMU snapshot interface for the TBU, request a snapshot.
tbu_ras_irpt[62]	Signal	Master	The RAS interrupt pin for errors detected in the TBUs.
tbu_reset_in[62]	Signal	Slave	Reset signals The TBUs can have independent reset signals. However, TCU reset will result in the reset of all TBUs. This behavior is unlike SMMU RTL implementations which do allow for independent reset of the TCU separate from TBUs. Each signal tbu_reset_in[n] corresponds to the TBU using tbs_pvbus_s[n]/tbs_pvbus_s[n] pair. If the SMMU receives a transaction whilst the TBU is expected to be in reset then it will complain using the ArchMsg.Warning.warning trace source. Those tbu_reset_in that correspond to a PCIe-RC connection can be connected to monitor the PCIe-RC's reset signal. If it receives an ATS request when in reset then it will complain in a similar way. You must connect these pins if you wish the TCU_NODE_STATUS for the nodes to be accurate (including any connected to the PCIe-RC).
tcu_pmu_irpt	Signal	Master	TCU Performance Monitoring Unit interrupt
tcu_pmusnapshot_ack	Signal	Master	PMU snapshot interface for the TCU, ack a snapshot.
tcu_pmusnapshot_req	Signal	Slave	PMU snapshot interface This is a four-phase handshake to have the corresponding PMCG perform a capture of the current counter values. PMU snapshot interface for the TCU, request a snapshot.
tcu_ras_irpt	Signal	Master	The RAS interrupt pin for errors detected in the TCU.
tcu_reset_in	Signal	Slave	The reset signal to the TCU interface.

## Parameters for MMU\_600

### TCUCFG\_XLATE\_SLOTS

#### Type

uint32\_t

#### Default value

512

Maximum number of outstanding stalled transactions that the SMMU supports.



Note

TCUCFG\_XLATE\_SLOTS must be  $\geq$  TCUCFG\_PTW\_SLOTS which is currently fixed to 512.

Accepted Values: 512

### all\_error\_messages\_through\_trace

#### Type

bool

**Default value**

false

Some conditions in the SMMU are so strange that the software programming the SMMU has done something wrong. At this point messages are output to either `ArchMsg.Error.*` or `ArchMsg.Warning.*` or to the error stream of the simulator. Outputting to the error stream of the simulator may cause it to return with a non-zero exit status.

If you set this option to true then instead of using the error stream of the simulator it will always use a trace stream allowing the simulation to exit with a zero exit status.

**behaviour\_of\_sampled\_at\_reset\_signals****Type**

unsigned

**Default value**

0

Some configuration signals into the SMMU are sampled on negedge of reset.

However, it can sometimes be hard to arrange to drive a configuration pin before the negedge of reset.

The configuration pins are sampled:

**0**

at negedge reset.

**1**

at negedge reset, but if a later change occurs at the same simulated time, and no transactions have occurred, then they will be resampled and the SMMU reset again.

**cmdq\_max\_number\_of\_commands\_to\_buffer****Type**

uint32\_t

**Default value**

10

The command queues can buffer fetched commands before issuing them. This parameter is roughly the maximum number of commands to do this for. The programmer visible effects are that just because the CONS pointer shows a command has been consumed does not necessarily mean that it has been issued (and completed). Higher values accentuate this effect.

**enable\_device\_id\_checks****Type**

bool

**Default value**

true

If this parameter is true then the DeviceIDs seen by the GIC are:

- **for client devices**

```
DeviceID = StreamID + translated_device_id_base
```

- **for SMMU-generated MSIs**

```
smmu_msi_device_id
```

This parameter enables two checks:

- If the DeviceID is used in the `output_attribute_transform` parameter, if it overflows 32 bits then the model warns. If the DeviceID is not used, it is assumed that the external agent that forms the DeviceID warns if it overflows.
- If the SMMU supports MSIs, the model checks that the GIC is able to distinguish an MSI generated by the SMMU from one generated by a client device.

As the exact mechanism to determine the DeviceID is in the system and not necessarily under control of the SMMU then you can disable these warnings using this parameter.

See also the parameters: `output_attribute_transform` and `msi_attribute_transform`.

**howto\_identify****Type**

string

**Default value**

"use-identify"

If `use-identify` then the SMMU uses the `identify` port to determine the `ssid`, `streamid`, `substreamid`. Otherwise, this string extracts them from the transaction's attributes.

Examples:

```
SEC_SID=ExtendedID[63], SSV=ExtendedID[62], SubstreamID=ExtendedID[51:32],  
StreamID=ExtendedID[31:0]
```

or

```
nSEC_SID=ExtendedID[63], StreamID=ExtendedID[55:24], nSSV=ExtendedID[20],  
SubstreamID=ExtendedID[19:0]  
StreamID[31:24]=0, StreamID[23:0]=ExtendedID[23:0], SSV=1[0], ...
```



Note

If you are not using the `use-identify` option then the configuration string is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

LHS Symbols:

**SIDV**

Indicates that the StreamID is valid.

**SSV**

Indicates that the SubstreamID is valid.

**SEC\_SID / SEC\_SID\_bit\_1**

Bits 0 and 1 respectively of the StreamID Security State

**StreamID**

(32 b) valid if `SIDV` is 1 or both `SIDV` and `NSIDV` are unused.

**SubstreamID**

(20 b) is valid if `SSV` is true.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

RHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming PVBUS transaction attributes

`SEC_SID` is 1b in the model. For RME systems, in hardware, then `SEC_SID` is 2b, in the model `SEC_SID_bit_1` represents bit[1].

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

Negative and positive logic symbols for the same attribute is an error.

The model uses the term SSD (Security State Determination) to mean which security state the transaction, register, structure, event, etc. belongs to.

In the SMMUV3 Architecture, the side-band signal `SEC_SID` holds the information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one bit signal. With RME, in the hardware, it was expanded to 2b. To retain backwards compatibility in the model, `SEC_SID` remains as 1b in this parameter, but the second bit can be expressed with `SEC_SID_bit_1` (and its negative logic version `nSEC_SID_bit_1`)

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0



For those systems that support NoStreamID transactions, and `howto_identify` is not using the port, first the SMMU determines the value of `sidv` or `nsidv` to see if the transaction is a NoStreamID transaction (`sidv == 0` or `nsidv == 1`). If so then the rest of the `howto_identify` string is ignored as the information extracted relates only to StreamID transactions. Instead more information is extracted using the parameter `howto_identify_NoStreamID_extra_info`.

In AMBA systems, the equivalent signal to `sidv` is called either `ARMMUVALID` or `AWMMUVALID`. Collectively they are known as `AxMMUVALID`.

### **`list_of_ns_sid_high_at_bitpos0`**

#### **Type**

string

#### **Default value**

`()`

A comma-separated list of values to bitwise OR into each Non-secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems.

The empty string corresponds to all 0s.

### **`list_of_pcie_mode`**

#### **Type**

string

#### **Default value**

`()`

A comma-separated list of ranges of ports that represent TBUs that are attached to PCIe Root Complexes (PCIe-RC). A single PCIe-RC might use several TBUs and stripe accesses across them. The attribute handling for these TBUs is slightly different in that if the PCIe transaction is NoSnoop and the output attributes of the translation would be weaker than `inc-osh` then the output is forced to `inc-osh`.

`inc-osh == "inner normal non-cacheable, out normal non-cacheable, outer shared"`

### **`list_of_pcie_rc`**

#### **Type**

string

#### **Default value**

`()`

This is a list of ports that are connected to PCIe Root Complex (PCIe-RC) by a protocol called DTI-ATS. This port is used to transport ATS and PRI Requests to the SMMU from the PCIe-RC.

In the real hardware, the PCIe-RC uses this port for ATS/PRI, and the actual transactions go through separate TBUs. In the model, this port can accept actual transactions as well. However, in the model, then the ATC Invalidates and the PRI Responses need to be transferred over the corresponding `pvbus_id_routed_m` port as DTI-ATS is bidirectional, but PVBUS is not.

### **`list_of_s_sid_high_at_bitpos0`**

#### **Type**

string

#### **Default value**

""

A comma-separated list of values to bitwise OR into each Secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

The empty string corresponds to all 0s.

`use_ns` can be used to apply the `list_of_ns_sid_high_at_bitpos0` values instead.

### **`msi_attribute_transform`**

#### **Type**

string

#### **Default value**

"ExtendedID[31:0]=smmu\_msi\_device\_id, ManagerID64[31:0]=0xFFFFffff"

Transform downstream attributes of MSI transactions.



**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter output attributes of SMMU-generated MSIs. Example:

```
"UserFlags[15:0]=smmu_msi_device_id[31:16],
ManagerID64[15:0]=smmu_msi_device_id[15:0],
ExtendedID=0"
```

LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Outgoing PVBUS transaction attributes

RHS Symbols:

**smmu\_msi\_device\_id**

The value stored in the parameter with the same name

**interrupt\_kind**

The selected bit corresponds to the interrupts listed below

**0/1**

EVENTQ s/ns

**2**

PRIQ

**3/4**

CMD\_SYNC s/ns

**5/6**

GERROR s/ns

**7/8**

PMCG s/ns

**9/10/11**

RAS FHI/ERI/CRI

**12/13**

gpf\_far/gpt\_cfg\_far

**14/15/16/17**

Realm EVENTQ/PRIQ/CMDQ/GERROR

**HWATTR\_KIND\_0**

PBHA information

**Numeric Literals**

Any number. Ex: 0x1234

ExtendedID/ManagerID64/MasterID/UserFlags start with values {0, 0xFFFFffff, 0xFFFFffff, 0} respectively.

Any bits with no transform are unchanged.

This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices.



**Note**

See also `output_attribute_transform` and `enable_device_id_checks`.

---



After 11.25 the `interrupt_kind` field was extended to 5 bits. This is strictly a non-backwards compatible change. However, the original 3 bits were insufficient to express all the interrupt kinds that exist.

## **number\_of\_ports**

### **Type**

unsigned

### **Default value**

1

The number of port pairs that the SMMU has.

## **output\_attribute\_transform**

### **Type**

string

### **Default value**

"ExtendedID[31:0]=DeviceID"

Transform the downstream attributes of a translated transaction.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID,
ManagerID64[10]=ManagerID64[11], ManagerID64[11]=ManagerID64[10]"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**DeviceID**

StreamID + translated\_device\_id\_base

**StreamID / SubstreamID / SEC\_SID / SSV**

Architectural information. See parameter `howto_identify` for more information.

**nSEC\_SID / nSSV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

**st1PBHA / st2PBHA**

Page Based Hardware Attributes from leaf descriptors (zero if unused).

**STE\_IMPDEF1**

STE[127:116]

**Numeric Literals**

Any number. Ex: 0x1234

The streamID has had ns\_sid\_high/s\_sid\_high ORred into it for the appropriate TBU.

**output\_id\_routed\_transform****Type**

string

**Default value**

"Address[27:12]=StreamID[15:0], PAS=SSD"



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

The SMMU generates the following ID-routed transaction on the pvbus\_id\_routed\_m bus:

- ATC Invalidate
- PRI Response

This parameter controls how the SMMU should express:

- the StreamID
- the Trusted (T) bit

The value is a comma-separated list of assignments:

```
Address[27:12]=StreamID[15:0], ExtendedID[60]=T, ExtendedID[15:0]=StreamID[31:16]
```

Address bits[11:0] cannot be used.

The LHS can be one of:

- PAS
- MasterID / ManagerID64 / ExtendedID / UserFlags
- Address

The RHS can be one of:

- a numeric constant
- `SSD`
- `T` or negative version `nT`
- `StreamID`

For realm (or 'Trusted') transactions, then `ssd=0b11`, `T=1`, `nT=0`. For non-secure (or 'Non-Trusted') transactions, then `ssd=0b01`, `T=0`, `nT=1`.

## **prefetch\_only\_requests**

### **Type**

unsigned

### **Default value**

0

The simulator supports 'prefetch-only' DMI requests, which can occur at any time and for any reason and are intended to be invisible to the end execution of the model and to the user.

**0**

deny all prefetch-only requests

**1**

- use debug requests for any page table walks
  - form and use debug TLB/cache entries
  - any faults will not record, but deny the prefetch request

**2**

- treat prefetch-only requests like normal transactions
  - use normal page table walk transactions
  - use and form normal TLB/cache entries
  - faults will alter the programmer-visible state of the SMMU

0 is the safest.

1 treats the access like a debug request and requires that debug page table walks are treated correctly downstream. Any descriptors that need HTTU to allow the transaction to proceed will fail the request.

2 is dangerous, it uses real transactions and reports faults that are unphysical. Real transactions can be `wait()`ed and this disobeys the SystemC spec for `get_direct_mem_ptr()`.

## **sec\_override**

### **Type**

bool

**Default value**

false

The IMP DEF port `sec_override` controls whether some of the registers are accessible to secure or non-secure transactions. This parameter is the default value assumed for that port if the port is not driven by a signal.

**seed****Type**

uint32\_t

**Default value**

0x12345678

Used to seed the pseudo-random number generator that the SMMU model uses.

**size\_of\_cd\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding CD structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_l1cd\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding L1CD descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_l1ste\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding L1STE descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_ste\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding STE structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_tlb****Type**

uint32\_t

**Default value**

0

The number of entries in the TLB. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**smmu\_msi\_device\_id****Type**

uint32\_t

**Default value**

0

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID.

See parameters `msi_attribute_transform` and `enable_device_id_checks`.

**sup\_btm****Type**

bool

**Default value**

true

The default value of the register field `SMMU_IDR0.BTM` and `SMMU_ROOT_IDR0.BGPTM` (when supported). This enables Broadcast TLB maintenance.

**sup\_cohacc****Type**

bool

**Default value**

true



The default value of the register `SMMU_IDR0.COHAAC`

### **sup\_oas**

#### **Type**

unsigned

#### **Default value**

5

The hardware has an input port `sup_oas[2:0]` that indicates what output address size (OAS) the system has. This is sampled at reset.

The model does not have this port as it is expected to be a constant for the system and not to change. Instead it is just a parameter.

The allowed values are:

**0**

32 bits

**1**

36 bits

**2**

40 bits

**3**

42 bits

**4**

44 bits

**5**

48 bits

### **sup\_sev**

#### **Type**

bool

#### **Default value**

true

The default value of the register `SMMU_IDR0.SEV`

### **tlb\_when\_do\_f\_tlb\_conflict\_on\_overlap**

#### **Type**

unsigned

#### **Default value**

0

If a TLB entry is created by a walk and it overlaps an existing entry, there are some architectural situations where the result is known. For all others, then an implementation is allowed to use an **UNPREDICTABLE** combination of the two entries, or it can generate `F_TLB_CONFLICT`:

- 0**  
never generate
- 1**  
sometimes generate
- 2**  
always generate

Conflicts between global and non-global entries are not detected by the model.

### **translated\_device\_id\_base**

#### **Type**

uint32\_t

#### **Default value**

0

When appropriately enabled, assume that client device accesses are translated to a DeviceID as seen by the GIC of:

```
StreamID + translated_device_id_base
```

See parameter `output_attribute_transform` and `enable_device_id_checks`.

### **tw\_qs\_attribute\_transform**

#### **Type**

string

#### **Default value**

""

Transform downstream attributes of table walk and queue transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**HWATTR\_KIND\_0**

PBHA information

**kind**

Transaction kind

- For a read:

**0/1**

L1STE/STE

**2/3**

L1CD/CD

**4/5**

S1/S2 TTD (including CAS)

**6**

CMDQ

**7**

VMS

**11/12**

LOGPT/L1GPT

**13/14**

LODPT/L1DPT

- For a write:

**0**

EVENTQ

**1**

PRIQ

ExtendedID/ManagerID64/MasterID/UserFlags start with values {0, 0xFFFFffff, 0xFFFFffff, 0} respectively.

Any bits with no transform are unchanged.



**Note**

See also `output_attribute_transform` and `msi_attribute_transform`.

---

**version****Type**

string

**Default value**

"r0p0"

The version of this product

**wait\_cmdq\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing something on the command queue. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffffff)) - 1]$ .

**wait\_eventq\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing something on the event queue. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffffff)) - 1]$ .

**wait\_imp\_def\_work\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing an IMP DEF operation. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffffff)) - 1]$ .

The IMP DEF work in this case is the number of ticks between raising pmusnapshot\_req and pmusnapshot\_ack being raised, and the converse operation.

**wait\_misc\_async\_actions\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing an async action that could be delayed is performed. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

**wait\_msi\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before sending an MSI. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

**wait\_pri\_req\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before processing a PRI Request. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

**wait\_pri\_resp\_ticks****Type**

uint64\_t

**Default value**

1

This is the time to wait before sending a PRI Response back to the PCIe subsystem. When a PRI Response is an auto-response then the ATC might immediately make a new ATS request, that immediately fails, immediately makes a PRI Request, or auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

### 3.10.63 MMU\_700

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1096: IP revisions support**

Revision	Quality level
r0p0	Full support
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Limitations

- No power control
- AMBA stash operations, destructive read and destructive hint operations are not supported on PVBUS and also are not supported by the device.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMCB_CEBID0` fields. The PMU is intended for demonstration purposes only and for driver development.



Between 11.17 and 11.18, the point of triggering for events changed for 'TLB miss' and this might lead to an (architecturally valid) change in the values captured in some circumstances.

- Limited RAS support
- The SYSCO interface is not implemented
- The low power interface is not implemented
- The IMP DEF MPAM register file is not implemented. This controls how the internal resources of the MMU-700 are partitioned.
- `tcu_sid[31:0]` is not modelled, instead the parameter `smmu_msi_device_id` is used.
- The HWATTR side-band signal is available by configuring `output_attribute_transform`, `msi_attribute_transform` and `tw_qs_attribute_transform` with `HWATTR_KIND_0`. For some transactions HWATTR comes from the `SMMU_S_AGBPA[3:0]`. In the hardware, this register has an 'Update' bit[31] that should be written as 1 and will be turned to zero when all transactions using the old value have completed. The model does not implement this behavior and the Update bit is **RAZ/WI**.
- Any configuration parameter listed in the TRM but not shown in this file is not supported.
- `TCU_STATUS.GNT_XLATE_SLOTS` always reads at 512

## Notes

- A bad configuration renders the model inactive.
- Some configurations can be adjusted by configuration pins. These are only sampled at the negative edge of the reset pin. If you want to use these pins then you must drive them before sending a negative edge on the reset pin. During simulation reset the component driving them must also drive this transition again.
- Debug reads to the registers do not disturb the state.

- Writes to registers with Update flags, including debug writes, are ignored if the Update flag is already set to one.
- Debug and real accesses to the registers must be 32 bits or 64 bits.
- The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping used is remembered by the bus infrastructure and is used for subsequent sufficiently similar transactions without requiring intervention from the SMMU model, and is not traced, for instance.

The range of addresses that the mapping is valid for is determined by the SMMU model, depending on architectural and model-implementation details. However, as it is unaware of any sign-extension or the mapping that the SoC does, the SoC is responsible for subsequently narrowing the range of addresses for which this mapping is valid. Typically, this is done automatically when using PVBUSMapper.

- The hardware is a distributed SMMU and is divided into:
  - A single Translation Control Unit (TCU)
    - Has a port for the programming interface of the SMMU
    - Receives DVM messages
    - Does all the page walking, queue manipulation, etc.
  - One or more Translation Bus Units (TBUs)
    - Translate transactions from upstream (client) device into downstream transactions.
  - Zero or more connections to PCIe Root Complexes (PCIe-RCs)
  - There can be a total of 62 TBUs and PCIe-RCs attached.
  - An interconnect connecting the TBUs, PCIe-RCs to the TCU.
- A PCIe-RC has one connection to the TCU to make ATS requests but the PCIe-RC uses one or more TBUs to transform the transactions and pass them to the memory system. In the model, those TBUs are configured by the parameter `list_of_pcie_mode`. The SMMU does not know which TBUs a particular PCIe-RC is attached to.
- The TBU ORs a value into each StreamID that it receives. In the model, this is configured by the parameters:
  - `list_of_s_sid_high_at_bitpos0`
  - `list_of_ns_sid_high_at_bitpos0`
- The TCU, TBU, and the interconnect are all represented by this single model component.
- In the model, a pair of ports `tbs_pvbus_s[i]/tbm_pvbus_m[i]` represent a TBU 'i' or the `tbs_pvbus_s[i]` represents an incoming connection for a PCIe-RC. The corresponding reverse connection from the TCU to the PCIe-RC is by a special bus called `pvbus_id_routed_m` that is used to transport ATC Invalidates to the PCIe-RC.
  - In order to reduce system construction complexity the `tbs_pvbus_s[i]/tbm_pvbus_m[i]` also acts as a TBU so that the PCIe-RC need not separate its normal transactions and its ATS requests.
  - However, ATC Invalidates are only sent to a port which appears in `list_of_pcie_rc`. It should be uniquely decoded to a single port based on `list_of_ns_sid_high_at_bitpos0`

and those ATC Invalidates must be routed to the correct PCIe subsystem in order to invalidate the cache of ATS Response in the subsystem. Thus all TBUs that a PCIe-RC uses must have a unique reverse mapping from stream id to port.

- The pin `sup_oas` is not supported, instead it is a parameter as it is assumed that it would be tied to a fixed value in any specific platform.
- The hardware only has a single cacheability attribute for input transactions, but PVBUS transports both inner and outer cacheability.
- For non-PCIe-mode TBUs (i.e. their index does not appear in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - For non-cache maintenance operations:
    - If the input attribute is any type of device then it is well defined as being outer-shared and Device-nGnRnE or Device-nGnRE. There is no support for Gathering or Reordering.
    - If the outer cacheable input attribute is normal then if it is Write-back then this is converted to Inner Write-back Outer Write-back (iWB-oWB) with the desired shareability. No Transient hint is supported and is always treated as non-transient.
    - This leaves all other normal memory types that are mapped to Inner Normal Non-cacheable, Outer Normal Non-cacheable outer shared (iNC-oNC-osh).
- Thus the upstream devices must present the cacheability in the *outer* cacheability attribute on PVBUS if it is cacheable. If it is a device type then both the inner and outer attributes must be set to the same. If it is iNC-oNC-osh then it must be presented as such.
- For PCIe-mode TBUs (i.e. their index appears in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - Input transactions are from PCIe and the only indication of the memory type is in the NoSnoop bit of the transaction. No shareability is transported.
    - NoSnoop interpreted as iNC-oNC-osh
    - ! NoSnoop interpreted as iWB-oWB-ish (note inner shareable)
  - If a NoSnoop transaction has an attribute transform applied to it and the result of the transform is weaker than iNC-oNC-osh then it is forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it is forced to iNC-oNC-osh. However, if the page tables transformed it to a device type then, as all device types are stronger than iNC-oNC-osh then it exits the SMMU as the device type.
  - In the model, transactions are classified by their incoming memory attributes as to whether they are NoSnoop or not and then normalized appropriately:
    - iWB-oWB-any-shareability transactions are interpreted as ! NoSnoop and therefore are normalized to iWB-oWB-ish.
    - Anything else is considered NoSnoop and therefore is normalized to iNC-oNC-osh.
  - Translated accesses also suffer the same interpretation to determine NoSnoop and how they are normalized. Thus they could enter the system with attributes different to if they were Untranslated Accesses translated by normal means.
  - It is expected that there is a component downstream of the SMMU that is aware of the system address map and will override the memory type to 'device' for any transaction that accesses a peripheral.



- The hardware has a single cacheability on input and, for transactions that are neither cache-maintenance operations nor PCIe transactions, normalizes the input to an architectural form before performing the SMMUv3 architectural transform:
  - Any device type is left untouched (the input can only represent Device-nGnRE and Device-nGnRnE).
  - If the input is Write-back (WB) then it is normalized to iWB-oWB with the incoming shareability.
  - If the input is anything else it is normalized to iNC-oNC-osh.
- The model accepts full architectural attributes of two levels of cacheability and so has to decide how to interpret this in terms of the hardware. For transactions that are not cache maintenance operations, the model replicates the outer attribute into the inner attribute and then performs the normalization that the hardware does.
- The hardware normalizes the architectural output attributes and outputs a single level of cacheability and a user flag (OC) specifying if the architectural attributes were cacheable in the outer cacheable domain. If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. If the original transaction was NoSnoop, then any weaker memory type is strengthened to iNC-oNC-osh so apply the following transform:

```

if      iWB-oWB-nsh/ish/osh      then output WB-nsh/ish/osh, OC = 1
else if i (NC/WT/WB) -o (WB/WT)  then output NC-Sys,      OC = 1
else if i (NC/WT/WB) -oNC        then output NC-Sys,      OC = 0
else if Device-(GRE/nGRE/nGnRE) then output DV-Sys,      OC = 0
else                               output SO-Sys,        OC = 0

```

- The model only normalizes according to PCIe but otherwise leaves the architectural attributes intact on the output bus.
- MSIs are issued on the `qtw_pvbus_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `MasterID=0xFFFFffff`.

In the hardware, there is no way of distinguishing Event Queue writes from MSI writes, however, this provides a mechanism that if the model system needs to distinguish then it can.

## Parameters for parsing transaction attributes

Some of the parameters are strings that share a common parsing format for extracting from and placing information into the transaction's attributes.

They can extract and place information into the following fields of a transaction's attributes:

- `ManagerID64` (64 bits)
- `MasterID` (32 bits) (This is the lower 32 bits of `ManagerID64`)
- `ExtendedID` (64 bits)
- `UserFlags` (32 bits)

The string parameter is parsed as a comma-separated list of:

```
lhs_expression=rhs_expression
```

The `lhs_expression` and `rhs_expression` can be an entire symbol, for example:

```
StreamID
```

or a bit, or bit slice:

```
StreamID[16]  
StreamID[31:20]
```

In `rhs_expression`, numeric constants (or slices of numeric constants) are also allowed:

```
StreamID[16]=1  
StreamID[16]=10[2]
```

A single symbol might be assigned from multiple non-contiguous arrays of bits from a mix of different RHS symbols:

```
StreamID[16]=1, StreamID[31:30]=ExtendedID[1:0],  
StreamID[15:0]=UserFlags[31:16], ...
```

In those cases where a left hand symbol can also appear on the right hand side, it is possible to swap bits and transform the symbol. For example, the following expression swaps bits 0 and 1 of the `ExtendedID`:

```
ExtendedID[0]=ExtendedID[1], ExtendedID[1]=ExtendedID[0]
```

Any bits of the attributes that have no transform specified are retained from the input.

The `lhs_expression` and `rhs_expression` must have the same bit width.

## Iris and MTI instances for MMU\_700

This model has the following Iris instances:

**Table 3-1097: MMU\_700 Iris instances**

InstanceName	ComponentName
MMU_700	MMU_700
MMU_700.register_file[0]	PVBusSlave
MMU_700.service_request_tbu[0]	PVBusSlave
MMU_700.service_request_tbu[10]	PVBusSlave
MMU_700.service_request_tbu[11]	PVBusSlave
MMU_700.service_request_tbu[12]	PVBusSlave

InstanceName	ComponentName
MMU_700.service_request_tbu[13]	PVBusSlave
MMU_700.service_request_tbu[14]	PVBusSlave
MMU_700.service_request_tbu[15]	PVBusSlave
MMU_700.service_request_tbu[16]	PVBusSlave
MMU_700.service_request_tbu[17]	PVBusSlave
MMU_700.service_request_tbu[18]	PVBusSlave
MMU_700.service_request_tbu[19]	PVBusSlave
MMU_700.service_request_tbu[1]	PVBusSlave
MMU_700.service_request_tbu[20]	PVBusSlave
MMU_700.service_request_tbu[21]	PVBusSlave
MMU_700.service_request_tbu[22]	PVBusSlave
MMU_700.service_request_tbu[23]	PVBusSlave
MMU_700.service_request_tbu[24]	PVBusSlave
MMU_700.service_request_tbu[25]	PVBusSlave
MMU_700.service_request_tbu[26]	PVBusSlave
MMU_700.service_request_tbu[27]	PVBusSlave
MMU_700.service_request_tbu[28]	PVBusSlave
MMU_700.service_request_tbu[29]	PVBusSlave
MMU_700.service_request_tbu[2]	PVBusSlave
MMU_700.service_request_tbu[30]	PVBusSlave
MMU_700.service_request_tbu[31]	PVBusSlave
MMU_700.service_request_tbu[32]	PVBusSlave
MMU_700.service_request_tbu[33]	PVBusSlave
MMU_700.service_request_tbu[34]	PVBusSlave
MMU_700.service_request_tbu[35]	PVBusSlave
MMU_700.service_request_tbu[36]	PVBusSlave
MMU_700.service_request_tbu[37]	PVBusSlave
MMU_700.service_request_tbu[38]	PVBusSlave
MMU_700.service_request_tbu[39]	PVBusSlave
MMU_700.service_request_tbu[3]	PVBusSlave
MMU_700.service_request_tbu[40]	PVBusSlave
MMU_700.service_request_tbu[41]	PVBusSlave
MMU_700.service_request_tbu[42]	PVBusSlave
MMU_700.service_request_tbu[43]	PVBusSlave
MMU_700.service_request_tbu[44]	PVBusSlave
MMU_700.service_request_tbu[45]	PVBusSlave
MMU_700.service_request_tbu[46]	PVBusSlave
MMU_700.service_request_tbu[47]	PVBusSlave
MMU_700.service_request_tbu[48]	PVBusSlave

InstanceName	ComponentName
MMU_700.service_request_tbu[49]	PVBusSlave
MMU_700.service_request_tbu[4]	PVBusSlave
MMU_700.service_request_tbu[50]	PVBusSlave
MMU_700.service_request_tbu[51]	PVBusSlave
MMU_700.service_request_tbu[52]	PVBusSlave
MMU_700.service_request_tbu[53]	PVBusSlave
MMU_700.service_request_tbu[54]	PVBusSlave
MMU_700.service_request_tbu[55]	PVBusSlave
MMU_700.service_request_tbu[56]	PVBusSlave
MMU_700.service_request_tbu[57]	PVBusSlave
MMU_700.service_request_tbu[58]	PVBusSlave
MMU_700.service_request_tbu[59]	PVBusSlave
MMU_700.service_request_tbu[5]	PVBusSlave
MMU_700.service_request_tbu[60]	PVBusSlave
MMU_700.service_request_tbu[61]	PVBusSlave
MMU_700.service_request_tbu[62]	PVBusSlave
MMU_700.service_request_tbu[63]	PVBusSlave
MMU_700.service_request_tbu[6]	PVBusSlave
MMU_700.service_request_tbu[7]	PVBusSlave
MMU_700.service_request_tbu[8]	PVBusSlave
MMU_700.service_request_tbu[9]	PVBusSlave
MMU_700.tbu[0]	PVBusMapper
MMU_700.tbu[10]	PVBusMapper
MMU_700.tbu[11]	PVBusMapper
MMU_700.tbu[12]	PVBusMapper
MMU_700.tbu[13]	PVBusMapper
MMU_700.tbu[14]	PVBusMapper
MMU_700.tbu[15]	PVBusMapper
MMU_700.tbu[16]	PVBusMapper
MMU_700.tbu[17]	PVBusMapper
MMU_700.tbu[18]	PVBusMapper
MMU_700.tbu[19]	PVBusMapper
MMU_700.tbu[1]	PVBusMapper
MMU_700.tbu[20]	PVBusMapper
MMU_700.tbu[21]	PVBusMapper
MMU_700.tbu[22]	PVBusMapper
MMU_700.tbu[23]	PVBusMapper
MMU_700.tbu[24]	PVBusMapper
MMU_700.tbu[25]	PVBusMapper

InstanceName	ComponentName
MMU_700.tbu[26]	PVBusMapper
MMU_700.tbu[27]	PVBusMapper
MMU_700.tbu[28]	PVBusMapper
MMU_700.tbu[29]	PVBusMapper
MMU_700.tbu[2]	PVBusMapper
MMU_700.tbu[30]	PVBusMapper
MMU_700.tbu[31]	PVBusMapper
MMU_700.tbu[32]	PVBusMapper
MMU_700.tbu[33]	PVBusMapper
MMU_700.tbu[34]	PVBusMapper
MMU_700.tbu[35]	PVBusMapper
MMU_700.tbu[36]	PVBusMapper
MMU_700.tbu[37]	PVBusMapper
MMU_700.tbu[38]	PVBusMapper
MMU_700.tbu[39]	PVBusMapper
MMU_700.tbu[3]	PVBusMapper
MMU_700.tbu[40]	PVBusMapper
MMU_700.tbu[41]	PVBusMapper
MMU_700.tbu[42]	PVBusMapper
MMU_700.tbu[43]	PVBusMapper
MMU_700.tbu[44]	PVBusMapper
MMU_700.tbu[45]	PVBusMapper
MMU_700.tbu[46]	PVBusMapper
MMU_700.tbu[47]	PVBusMapper
MMU_700.tbu[48]	PVBusMapper
MMU_700.tbu[49]	PVBusMapper
MMU_700.tbu[4]	PVBusMapper
MMU_700.tbu[50]	PVBusMapper
MMU_700.tbu[51]	PVBusMapper
MMU_700.tbu[52]	PVBusMapper
MMU_700.tbu[53]	PVBusMapper
MMU_700.tbu[54]	PVBusMapper
MMU_700.tbu[55]	PVBusMapper
MMU_700.tbu[56]	PVBusMapper
MMU_700.tbu[57]	PVBusMapper
MMU_700.tbu[58]	PVBusMapper
MMU_700.tbu[59]	PVBusMapper
MMU_700.tbu[5]	PVBusMapper
MMU_700.tbu[60]	PVBusMapper

InstanceName	ComponentName
MMU_700.tbv[61]	PVBusMapper
MMU_700.tbv[62]	PVBusMapper
MMU_700.tbv[63]	PVBusMapper
MMU_700.tbv[6]	PVBusMapper
MMU_700.tbv[7]	PVBusMapper
MMU_700.tbv[8]	PVBusMapper
MMU_700.tbv[9]	PVBusMapper

This model has the following MTI trace components:

**Table 3-1098: MMU\_700 MTI instances**

InstanceName	ComponentName
MMU_700	MMU_700
MMU_700.register_file[0]	PVBusSlave
MMU_700.service_request_tbu[0]	PVBusSlave
MMU_700.service_request_tbu[10]	PVBusSlave
MMU_700.service_request_tbu[11]	PVBusSlave
MMU_700.service_request_tbu[12]	PVBusSlave
MMU_700.service_request_tbu[13]	PVBusSlave
MMU_700.service_request_tbu[14]	PVBusSlave
MMU_700.service_request_tbu[15]	PVBusSlave
MMU_700.service_request_tbu[16]	PVBusSlave
MMU_700.service_request_tbu[17]	PVBusSlave
MMU_700.service_request_tbu[18]	PVBusSlave
MMU_700.service_request_tbu[19]	PVBusSlave
MMU_700.service_request_tbu[1]	PVBusSlave
MMU_700.service_request_tbu[20]	PVBusSlave
MMU_700.service_request_tbu[21]	PVBusSlave
MMU_700.service_request_tbu[22]	PVBusSlave
MMU_700.service_request_tbu[23]	PVBusSlave
MMU_700.service_request_tbu[24]	PVBusSlave
MMU_700.service_request_tbu[25]	PVBusSlave
MMU_700.service_request_tbu[26]	PVBusSlave
MMU_700.service_request_tbu[27]	PVBusSlave
MMU_700.service_request_tbu[28]	PVBusSlave
MMU_700.service_request_tbu[29]	PVBusSlave
MMU_700.service_request_tbu[2]	PVBusSlave
MMU_700.service_request_tbu[30]	PVBusSlave
MMU_700.service_request_tbu[31]	PVBusSlave
MMU_700.service_request_tbu[32]	PVBusSlave

InstanceName	ComponentName
MMU_700.service_request_tbu[33]	PVBusSlave
MMU_700.service_request_tbu[34]	PVBusSlave
MMU_700.service_request_tbu[35]	PVBusSlave
MMU_700.service_request_tbu[36]	PVBusSlave
MMU_700.service_request_tbu[37]	PVBusSlave
MMU_700.service_request_tbu[38]	PVBusSlave
MMU_700.service_request_tbu[39]	PVBusSlave
MMU_700.service_request_tbu[3]	PVBusSlave
MMU_700.service_request_tbu[40]	PVBusSlave
MMU_700.service_request_tbu[41]	PVBusSlave
MMU_700.service_request_tbu[42]	PVBusSlave
MMU_700.service_request_tbu[43]	PVBusSlave
MMU_700.service_request_tbu[44]	PVBusSlave
MMU_700.service_request_tbu[45]	PVBusSlave
MMU_700.service_request_tbu[46]	PVBusSlave
MMU_700.service_request_tbu[47]	PVBusSlave
MMU_700.service_request_tbu[48]	PVBusSlave
MMU_700.service_request_tbu[49]	PVBusSlave
MMU_700.service_request_tbu[4]	PVBusSlave
MMU_700.service_request_tbu[50]	PVBusSlave
MMU_700.service_request_tbu[51]	PVBusSlave
MMU_700.service_request_tbu[52]	PVBusSlave
MMU_700.service_request_tbu[53]	PVBusSlave
MMU_700.service_request_tbu[54]	PVBusSlave
MMU_700.service_request_tbu[55]	PVBusSlave
MMU_700.service_request_tbu[56]	PVBusSlave
MMU_700.service_request_tbu[57]	PVBusSlave
MMU_700.service_request_tbu[58]	PVBusSlave
MMU_700.service_request_tbu[59]	PVBusSlave
MMU_700.service_request_tbu[5]	PVBusSlave
MMU_700.service_request_tbu[60]	PVBusSlave
MMU_700.service_request_tbu[61]	PVBusSlave
MMU_700.service_request_tbu[62]	PVBusSlave
MMU_700.service_request_tbu[63]	PVBusSlave
MMU_700.service_request_tbu[6]	PVBusSlave
MMU_700.service_request_tbu[7]	PVBusSlave
MMU_700.service_request_tbu[8]	PVBusSlave
MMU_700.service_request_tbu[9]	PVBusSlave
MMU_700.tbu[0]	PVBusMapper

InstanceName	ComponentName
MMU_700.tbu[10]	PVBusMapper
MMU_700.tbu[11]	PVBusMapper
MMU_700.tbu[12]	PVBusMapper
MMU_700.tbu[13]	PVBusMapper
MMU_700.tbu[14]	PVBusMapper
MMU_700.tbu[15]	PVBusMapper
MMU_700.tbu[16]	PVBusMapper
MMU_700.tbu[17]	PVBusMapper
MMU_700.tbu[18]	PVBusMapper
MMU_700.tbu[19]	PVBusMapper
MMU_700.tbu[1]	PVBusMapper
MMU_700.tbu[20]	PVBusMapper
MMU_700.tbu[21]	PVBusMapper
MMU_700.tbu[22]	PVBusMapper
MMU_700.tbu[23]	PVBusMapper
MMU_700.tbu[24]	PVBusMapper
MMU_700.tbu[25]	PVBusMapper
MMU_700.tbu[26]	PVBusMapper
MMU_700.tbu[27]	PVBusMapper
MMU_700.tbu[28]	PVBusMapper
MMU_700.tbu[29]	PVBusMapper
MMU_700.tbu[2]	PVBusMapper
MMU_700.tbu[30]	PVBusMapper
MMU_700.tbu[31]	PVBusMapper
MMU_700.tbu[32]	PVBusMapper
MMU_700.tbu[33]	PVBusMapper
MMU_700.tbu[34]	PVBusMapper
MMU_700.tbu[35]	PVBusMapper
MMU_700.tbu[36]	PVBusMapper
MMU_700.tbu[37]	PVBusMapper
MMU_700.tbu[38]	PVBusMapper
MMU_700.tbu[39]	PVBusMapper
MMU_700.tbu[3]	PVBusMapper
MMU_700.tbu[40]	PVBusMapper
MMU_700.tbu[41]	PVBusMapper
MMU_700.tbu[42]	PVBusMapper
MMU_700.tbu[43]	PVBusMapper
MMU_700.tbu[44]	PVBusMapper
MMU_700.tbu[45]	PVBusMapper



InstanceName	ComponentName
MMU_700.tbu[46]	PVBusMapper
MMU_700.tbu[47]	PVBusMapper
MMU_700.tbu[48]	PVBusMapper
MMU_700.tbu[49]	PVBusMapper
MMU_700.tbu[4]	PVBusMapper
MMU_700.tbu[50]	PVBusMapper
MMU_700.tbu[51]	PVBusMapper
MMU_700.tbu[52]	PVBusMapper
MMU_700.tbu[53]	PVBusMapper
MMU_700.tbu[54]	PVBusMapper
MMU_700.tbu[55]	PVBusMapper
MMU_700.tbu[56]	PVBusMapper
MMU_700.tbu[57]	PVBusMapper
MMU_700.tbu[58]	PVBusMapper
MMU_700.tbu[59]	PVBusMapper
MMU_700.tbu[5]	PVBusMapper
MMU_700.tbu[60]	PVBusMapper
MMU_700.tbu[61]	PVBusMapper
MMU_700.tbu[62]	PVBusMapper
MMU_700.tbu[63]	PVBusMapper
MMU_700.tbu[6]	PVBusMapper
MMU_700.tbu[7]	PVBusMapper
MMU_700.tbu[8]	PVBusMapper
MMU_700.tbu[9]	PVBusMapper

## Ports for MMU\_700

**Table 3-1099: Ports**

Name	Protocol	Type	Description
axi_stream_msi_addr_to_match_s	Value_64	Slave	Any SMMU-originated MSI which exactly matches the address in this port will be sent through the AXI stream port axi_stream_msi_m which is usually connected to the GIC through axi_stream_msi_s. As MSIs are 32 bit aligned then if bits[1:0] != 0 or the address is above OAS then this is effectively disabled. NOTE that the model does not support tcu_sid[31:0] which is the MSI DeviceID to send on axi_stream_msi_m. Instead the parameter smmu_msi_device_id is used. See also the parameters: axi_stream_msi_TID and axi_stream_msi_TDEST. The default value of this port is set by the parameter axi_stream_msi_addr_to_match. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
axi_stream_msi_m	PVBus	Master	Manager port used for sending SMMU originated MSIs directly to the GIC

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock signal (in RTL aclk) This is a clock time-base used by the TCU to spread some of its processing over time, if enabled by the wait_* parameters. The clock must always be connected.
cmd_sync_irpt_ns	Signal	Master	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of SIG_IRQ.
cmd_sync_irpt_s	Signal	Master	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
event_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
event_q_irpt_s	Signal	Master	Pulsed interrupt output signal for the secure event queue becoming non-empty.
evento	Signal	Master	Event signal
global_irpt_ns	Signal	Master	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signalling an error.
global_irpt_s	Signal	Master	Pulsed interrupt output signal for secure SMMU_S_GERROR(N) signalling an error.
identify	SMMUv3AEMIdentifyProtocol	Master	Map the transaction to the tuple (StreamID, SubStreamID, SubStreamIDValid, SSD) The StreamID that is produced by the implementation of this protocol is not the final StreamID. The final StreamID is produced by using the list_of_ns_sid_high_at_bitpos0/list_of_s_sid_high_at_bitpos0 parameter to map the StreamID based on the upstream port index. Also see the parameter howto_identify which can replace the functionality of this port under certain circumstances.
pri_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the PRI queue becoming non-empty.
prog_pvbus_s	PVBus	Slave	Register subordinate port (in RTL PROG)
pvbus_id_routed_m[62]	PVBus	Master	This is a special "id-routed" port for transmitting ATC invalidates upstream into the PCIe EndPoints, it is not a normal bus. See the parameter output_id_routed_transform. The FastModels ATC Invalidate and PRI Response protocol specifies how to route and deal with this port.
qtw_pvbus_m	PVBus	Master	Manager port used for Table Walks, MSIs and Queue access. Note that although this is a manager port, it can still receive DVM messages.
sec_override	Signal	Slave	Allow certain registers to be accessible to non-secure accesses from reset, as described in the TCU_SCR register. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_btm	Signal	Slave	System supports BTM and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If BTM (Broadcast Table Maintenance) is not supported then DVM messages will be ignored. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_cohacc	Signal	Slave	System supports COHACC and will be reflected in the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.

Name	Protocol	Type	Description
sup_httu	Signal	Slave	System supports HTTU and will be reflected in the value of SMMU_IDR0.HTTU: - 0b00 (HTTU not supported) if sup_httu is 0 - 0b10 (update of AF and Dirty flags supported) if sup_httu is 1 The default value for this pin is 1. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_sev	Signal	Slave	System supports SEV and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
tbm_pvbus_m[62]	PVBus	Master	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered tbs_pvbus_s[] port.
tbs_pvbus_s[62]	PVBus	Slave	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
tbu_cri_irpt[62]	Signal	Master	Critical error interrupt for RAS events from the TBU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE in the hardware then this is called ras_cri.
tbu_eri_irpt[62]	Signal	Master	Error Recovery Interrupt for RAS events from the TBU. This is used for uncorrected errors. NOTE in the hardware then this is called ras_eri.
tbu_fhi_irpt[62]	Signal	Master	Fault Handling Interrupt for RAS events from the TBU. Usually this is for corrected errors. NOTE in the hardware then this is called ras_fhi.
tbu_pmu_irpt[62]	Signal	Master	TBU Performance Monitoring Unit interrupt, one per TBU.
tbu_pmusnapshot_ack[62]	Signal	Master	PMU snapshot interface for the TBU, ack a snapshot.
tbu_pmusnapshot_req[62]	Signal	Slave	PMU snapshot interface for the TBU, request a snapshot.
tbu_reset_in[62]	Signal	Slave	Reset signals The TBUs can have independent reset signals. However, TCU reset will result in the reset of all TBUs. This behavior is unlike SMMU RTL implementations which do allow for independent reset of the TCU separate from TBUs. Each signal tbu_reset_in[n] corresponds to the TBU using tbs_pvbus_s[n]/tbs_pvbus_s[n] pair. If the SMMU receives a transaction whilst the TBU is expected to be in reset then it will complain using the ArchMsg.Warning.warning trace source. Those tbu_reset_in that correspond to a PCIe-RC connection can be connected to monitor the PCIe-RC's reset signal. If it receives an ATS request when in reset then it will complain in a similar way. You must connect these pins if you wish the TCU_NODE_STATUS for the nodes to be accurate (including any connected to the PCIe-RC).
tcu_cri_irpt	Signal	Master	Critical error interrupt for RAS events from the TCU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE in the hardware then this is called ras_cri.

Name	Protocol	Type	Description
tcu_eri_irpt	Signal	Master	Error Recovery Interrupt for RAS events from the TCU. This is used for uncorrected errors. NOTE in the hardware then this is called ras_eri.
tcu_fhi_irpt	Signal	Master	Fault Handling Interrupt for RAS events from the TCU. Usually this is for corrected errors. NOTE in the hardware then this is called ras_fhi.
tcu_pmu_irpt	Signal	Master	TCU Performance Monitoring Unit interrupt
tcu_pmusnapshot_ack	Signal	Master	PMU snapshot interface for the TCU, ack a snapshot.
tcu_pmusnapshot_req	Signal	Slave	PMU snapshot interface This is a four-phase handshake to have the corresponding PMCG perform a capture of the current counter values. PMU snapshot interface for the TCU, request a snapshot.
tcu_reset_in	Signal	Slave	The reset signal to the TCU interface.

## Parameters for MMU\_700

### TCUCFG\_PARTID\_WIDTH

#### Type

unsigned

#### Default value

9

The width of the MPAM PARTID on the bus.

See also parameter mpam\_attribute\_transform. Accepted Values: 1 6 9

### TCUCFG\_XLATE\_SLOTS

#### Type

uint32\_t

#### Default value

512

Maximum number of outstanding stalled transactions that the SMMU supports.



Note

TCUCFG\_XLATE\_SLOTS must be  $\geq$  TCUCFG\_PTW\_SLOTS which is currently fixed to 512.

Accepted Values: 512 1024 2048 4096

### all\_error\_messages\_through\_trace

#### Type

bool

**Default value**

false

Some conditions in the SMMU are so strange that the software programming the SMMU has done something wrong. At this point messages are output to either `ArchMsg.Error.*` or `ArchMsg.Warning.*` or to the error stream of the simulator. Outputting to the error stream of the simulator may cause it to return with a non-zero exit status.

If you set this option to true then instead of using the error stream of the simulator it will always use a trace stream allowing the simulation to exit with a zero exit status.

**axi\_stream\_msi\_TDEST****Type**

uint32\_t

**Default value**

0

ID of the AXI stream subordinate port on the GIC that receives SMMU originated MSIs sent directly. The name of the GIC port is `axi_stream_msi_s`.

**Note**

If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TID`

**axi\_stream\_msi\_TID****Type**

uint32\_t

**Default value**

0

ID of the AXI stream manager port that sends SMMU TCU originated MSIs directly to the GIC. The name of the SMMU port is `axi_stream_msi_m`.

**Note**

If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TDEST`

### **`axi_stream_msi_addr_to_match`**

#### **Type**

`uint64_t`

#### **Default value**

`0xFFFFFFFFFFFFFFFF`

If the last two bits are 0, any SMMU-originated MSI which exactly matches the address will be sent through the AXI stream port `axi_stream_msi_m` which is usually connected to the GIC.

This parameter drives the value of the `axi_stream_msi_addr_to_match_s` port at simulation reset. For every reset after that, the value of the port will be sampled and used if changed.



The entire address must match, including bits [1:0]. As MSIs are 32 bit aligned then if `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_TID`
- `axi_stream_msi_TDEST`

### **`behaviour_of_sampled_at_reset_signals`**

#### **Type**

unsigned

#### **Default value**

0

Some configuration signals into the SMMU are sampled on negedge of reset.

However, it can sometimes be hard to arrange to drive a configuration pin before the negedge of reset.

The configuration pins are sampled:

**0**

at negedge reset.

**1**

at negedge reset, but if a later change occurs at the same simulated time, and no transactions have occurred, then they will be resampled and the SMMU reset again.

**cmdq\_max\_number\_of\_commands\_to\_buffer****Type**

uint32\_t

**Default value**

10

The command queues can buffer fetched commands before issuing them. This parameter is roughly the maximum number of commands to do this for. The programmer visible effects are that just because the CONS pointer shows a command has been consumed does not necessarily mean that it has been issued (and completed). Higher values accentuate this effect.

**enable\_device\_id\_checks****Type**

bool

**Default value**

true

If this parameter is true then the DeviceIDs seen by the GIC are:

- **for client devices**

$$\text{DeviceID} = \text{StreamID} + \text{translated\_device\_id\_base}$$

- **for SMMU-generated MSIs**

$$\text{smmu\_msi\_device\_id}$$

This parameter enables two checks:

- If the DeviceID is used in the `output_attribute_transform` parameter, if it overflows 32 bits then the model warns. If the DeviceID is not used, it is assumed that the external agent that forms the DeviceID warns if it overflows.
- If the SMMU supports MSIs, the model checks that the GIC is able to distinguish an MSI generated by the SMMU from one generated by a client device.

As the exact mechanism to determine the DeviceID is in the system and not necessarily under control of the SMMU then you can disable these warnings using this parameter.

See also the parameters: `output_attribute_transform` and `msi_attribute_transform`.

**howto\_identify****Type**

string

**Default value**

"use-identify"

If `use-identify` then the SMMU uses the `identify` port to determine the `ssd`, `StreamID`, `SubStreamID`. Otherwise, this string extracts them from the transaction's attributes.

Examples:

```
SEC_SID=ExtendedID[63], SSV=ExtendedID[62], SubstreamID=ExtendedID[51:32],
StreamID=ExtendedID[31:0]
```

or

```
nSEC_SID=ExtendedID[63], StreamID=ExtendedID[55:24], nSSV=ExtendedID[20],
SubstreamID=ExtendedID[19:0]
StreamID[31:24]=0, StreamID[23:0]=ExtendedID[23:0], SSV=1[0], ...
```



Note

If you are not using the `use-identify` option then the configuration string is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

LHS Symbols:

**SIDV**

Indicates that the StreamID is valid.

**SSV**

Indicates that the SubstreamID is valid.

**SEC\_SID / SEC\_SID\_bit\_1**

Bits 0 and 1 respectively of the StreamID Security State

**StreamID**

(32 b) valid if `SIDV` is 1 or both `SIDV` and `nSIDV` are unused.

**SubstreamID**

(20 b) is valid if `ssv` is true.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

RHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming PVBUS transaction attributes

`SEC_SID` is 1b in the model. For RME systems, in hardware, then `SEC_SID` is 2b, in the model `SEC_SID_bit_1` represents bit[1].

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

Negative and positive logic symbols for the same attribute is an error.



The model uses the term SSD (Security State Determination) to mean which security state the transaction, register, structure, event, etc. belongs to.

In the SMMUv3 Architecture, the side-band signal `SEC_SID` holds the information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one bit signal. With RME, in the hardware, it was expanded to 2b. To retain backwards compatibility in the model, `SEC_SID` remains as 1b in this parameter, but the second bit can be expressed with `SEC_SID_bit_1` (and its negative logic version `nSEC_SID_bit_1`)

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

For those systems that support NoStreamID transactions, and `howto_identify` is not using the port, first the SMMU determines the value of `SIDV` or `nSIDV` to see if the transaction is a NoStreamID transaction (`SIDV == 0` or `nSIDV == 1`). If so then the rest of the `howto_identify` string is ignored as the information extracted relates only to StreamID transactions. Instead more information is extracted using the parameter `howto_identify_NoStreamID_extra_info`.

In AMBA systems, the equivalent signal to `SIDV` is called either `ARMMUVALID` OR `AWMMUVALID`. Collectively they are known as `AxMMUVALID`.

**ish\_is\_osh\_DANGER**


**Type**

bool


**Default value**

true

When set, any transaction that would use the architectural inner shareable domain is converted to use the outer shareable domain.

- 

Note

This parameter should match the equivalent `ish_is_osh` from the PE. If an incompatible value of the `ish_is_osh` parameter is configured for the PE and the SMMU, data coherency may be compromised.
- 

Note

All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

**list\_of\_ns\_sid\_high\_at\_bitpos0****Type**

string

**Default value**

""

A comma-separated list of values to bitwise OR into each Non-secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems.

The empty string corresponds to all 0s.

**list\_of\_pcie\_mode****Type**

string

**Default value**

""

A comma-separated list of ranges of ports that represent TBUs that are attached to PCIe Root Complexes (PCIe-RC). A single PCIe-RC might use several TBUs and stripe accesses across them. The attribute handling for these TBUs is slightly different in that if the PCIe transaction is NoSnoop and the output attributes of the translation would be weaker than `iNC-oNC-osh` then the output is forced to `iNC-oNC-osh`.

`iNC-oNC-osh` == "inner normal non-cacheable, out normal non-cacheable, outer shared"

**list\_of\_pcie\_rc****Type**

string

**Default value**

""

This is a list of ports that are connected to PCIe Root Complex (PCIe-RC) by a protocol called DTI-ATS. This port is used to transport ATS and PRI Requests to the SMMU from the PCIe-RC.

In the real hardware, the PCIe-RC uses this port for ATS/PRI, and the actual transactions go through separate TBUs. In the model, this port can accept actual transactions as well. However, in the model, then the ATC Invalidates and the PRI Responses need to be transferred over the corresponding `pvbus_id_routed_m` port as DTI-ATS is bidirectional, but PVBUS is not.

**list\_of\_s\_sid\_high\_at\_bitpos0****Type**

string

**Default value**

""

A comma-separated list of values to bitwise OR into each Secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

The empty string corresponds to all 0s.

use-ns can be used to apply the list\_of\_ns\_sid\_high\_at\_bitpos0 values instead.

**mpam\_attribute\_transform****Type**

string

**Default value**

"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID,  
ExtendedID[38]=MPAM\_NS"

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MPAM is supported, this is applied to *all* downstream transactions to transport the MPAM information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID,  
ExtendedID[38]=MPAM_NS"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/MasterID/UserFlags.

RHS Symbols:

- MPAM\_PARTID
- MPAM\_PMG
- MPAM\_NS

Any bits with no transform are unchanged.

**Note**

- attribute transforms applied before this:
  - for client transactions 'output\_attribute\_transform'.
  - for table walks `tw_qs_attribute_transform`.
  - for MSIs `msi_attribute_transform`.
- for translated transactions from client devices then `MPAM_NS = ! SEC_SID`.

## **msi\_attribute\_transform**

### Type

string

### Default value

"ExtendedID[31:0]=smmu\_msi\_device\_id, ManagerID64[31:0]=0xFFFFFFFF"

Transform downstream attributes of MSI transactions.

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter output attributes of SMMU-generated MSIs. Example:

```
"UserFlags[15:0]=smmu_msi_device_id[31:16],
  ManagerID64[15:0]=smmu_msi_device_id[15:0],
  ExtendedID=0"
```

LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Outgoing PVBUS transaction attributes

RHS Symbols:

**smmu\_msi\_device\_id**

The value stored in the parameter with the same name

**interrupt\_kind**

The selected bit corresponds to the interrupts listed below

**0/1**

EVENTQ s/ns

**2**

PRIQ

- 3/4  
CMD\_SYNC s/ns
- 5/6  
GERROR s/ns
- 7/8  
PMCG s/ns
- 9/10/11  
RAS FHI/ERI/CRI
- 12/13  
gpf\_far/gpt\_cfg\_far
- 14/15/16/17  
Realm EVENTQ/PRIQ/CMDQ/GERROR

**HWATTR\_KIND\_0**  
PBHA information

**Numeric Literals**  
Any number. Ex: 0x1234

ExtendedID/ManagerID64/MasterID/UserFlags start with values {0, 0xFFFFFFFF, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.

This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices.



See also `output_attribute_transform` and `enable_device_id_checks`.



After 11.25 the `interrupt_kind` field was extended to 5 bits. This is strictly a non-backwards compatible change. However, the original 3 bits were insufficient to express all the interrupt kinds that exist.

---

**`normalize_input_normal_non_iWB_oWB_to_iNC_oNC_osh_DANGER`**

Type

bool

Default value

true

When set, use Inner Non Cacheable, Outer Non Cacheable, Outer Shareable for any upstream transaction that would use any of the following attributes:

- Normal Non-cacheable Bufferable
- Normal Non-cacheable Non-bufferable
- Write-through



Note

This parameter should match the equivalent configuration from the PE. If an incompatible value of this parameter is configured for the PE and the SMMU, data coherency may be compromised.

All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

## number\_of\_ports

### Type

unsigned

### Default value

1

The number of port pairs that the SMMU has.

## output\_attribute\_transform

### Type

string

### Default value

"ExtendedID[31:0]=DeviceID"

Transform the downstream attributes of a translated transaction.



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID,
ManagerID64[10]=ManagerID64[11], ManagerID64[11]=ManagerID64[10]"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**  
Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**DeviceID**  
StreamID + translated\_device\_id\_base

**StreamID / SubstreamID / SEC\_SID / SSV**  
Architectural information. See parameter `howto_identify` for more information.

**nSEC\_SID / nSSV**  
Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

**St1PBHA / St2PBHA**  
Page Based Hardware Attributes from leaf descriptors (zero if unused).

**STE\_IMPDEF1**  
STE[127:116]

**HWATTR\_KIND\_0**  
PBHA information

**Numeric Literals**  
Any number. Ex: 0x1234

The `streamID` has had `ns_sid_high/s_sid_high` ORred into it for the appropriate TBU.



Note

‘mpam\_attribute\_transform’ is applied after this.

**output\_id\_routed\_transform**

**Type**  
string

**Default value**  
“Address[27:12]=StreamID[15:0], PAS=SSD”



Note

The parameter is parsed according to the rules in the section ‘Parameters for parsing transaction attributes’.

The SMMU generates the following ID-routed transaction on the `pvbus_id_routed_m` bus:

- ATC Invalidate
- PRI Response

This parameter controls how the SMMU should express:

- the StreamID
- the Trusted (T) bit

The value is a comma-separated list of assignments:

```
Address[27:12]=StreamID[15:0], ExtendedID[60]=T, ExtendedID[15:0]=StreamID[31:16]
```

Address bits[11:0] cannot be used.

The LHS can be one of:

- PAS
- MasterID / ManagerID64 / ExtendedID / UserFlags
- Address

The RHS can be one of:

- a numeric constant
- SSD
- T or negative version nT
- StreamID

For realm (or 'Trusted') transactions, then `ssd=0b11`, `T=1`, `nT=0`. For non-secure (or 'Non-Trusted') transactions, then `ssd=0b01`, `T=0`, `nT=1`.

## **prefetch\_only\_requests**

### **Type**

unsigned

### **Default value**

0

The simulator supports 'prefetch-only' DMI requests, which can occur at any time and for any reason and are intended to be invisible to the end execution of the model and to the user.

**0**

deny all prefetch-only requests

**1**

- use debug requests for any page table walks
  - form and use debug TLB/cache entries
  - any faults will not record, but deny the prefetch request

**2**

- treat prefetch-only requests like normal transactions



- use normal page table walk transactions
- use and form normal TLB/cache entries
- faults will alter the programmer-visible state of the SMMU

0 is the safest.

1 treats the access like a debug request and requires that debug page table walks are treated correctly downstream. Any descriptors that need HTTU to allow the transaction to proceed will fail the request.

2 is dangerous, it uses real transactions and reports faults that are unphysical. Real transactions can be `wait()`ed and this disobeys the SystemC spec for `get_direct_mem_ptr()`.

### **sec\_override**

#### **Type**

bool

#### **Default value**

false

The IMP DEF port `sec_override` controls whether some of the registers are accessible to secure or non-secure transactions. This parameter is the default value assumed for that port if the port is not driven by a signal.

### **seed**

#### **Type**

uint32\_t

#### **Default value**

0x12345678

Used to seed the pseudo-random number generator that the SMMU model uses.

### **size\_of\_cd\_cache**

#### **Type**

uint32\_t

#### **Default value**

0

The number of entries in the cache holding CD structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

### **size\_of\_l1cd\_cache**

#### **Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding L1CD descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_l1ste\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding L1STE descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_ste\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding STE structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_tlb****Type**

uint32\_t

**Default value**

0

The number of entries in the TLB. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**smmu\_msi\_device\_id****Type**

uint32\_t

**Default value**

0

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID.

See parameters `msi_attribute_transform` and `enable_device_id_checks`.

**sup\_btm****Type**

bool

**Default value**

true

The default value of the register field `SMMU_IDR0.BTM` and `SMMU_ROOT_IDR0.BGPTM` (when supported). This enables Broadcast TLB maintenance.

**sup\_cohacc****Type**

bool

**Default value**

true

The default value of the register `SMMU_IDR0.COHAAC`

**sup\_httu****Type**

bool

**Default value**

true

The initial value of the `sup_httu` port. See the port description for `sup_httu`.

**sup\_oas****Type**

unsigned

**Default value**

6

The hardware has an input port `sup_oas[2:0]` that indicates what output address size (OAS) the system has. This is sampled at reset.

The model does not have this port as it is expected to be a constant for the system and not to change. Instead it is just a parameter.

The allowed values are:

**0**

32 bits

**1**

36 bits

- 2  
40 bits
- 3  
42 bits
- 4  
44 bits
- 5  
48 bits
- 6  
52 bits

**sup\_sev**  
**Type**  
bool  
**Default value**  
true

The default value of the register `SMMU_IDR0.SEV`

**tlb\_when\_do\_f\_tlb\_conflict\_on\_overlap**  
**Type**  
unsigned  
**Default value**  
0

If a TLB entry is created by a walk and it overlaps an existing entry, there are some architectural situations where the result is known. For all others, then an implementation is allowed to use an **UNPREDICTABLE** combination of the two entries, or it can generate `F_TLB_CONFLICT`:

- 0  
never generate
- 1  
sometimes generate
- 2  
always generate

Conflicts between global and non-global entries are not detected by the model.

**translated\_device\_id\_base**  
**Type**  
uint32\_t

**Default value**

0

When appropriately enabled, assume that client device accesses are translated to a DeviceID as seen by the GIC of:

```
StreamID + translated_device_id_base
```

See parameter `output_attribute_transform` and `enable_device_id_checks`.

**tw\_qs\_attribute\_transform****Type**

string

**Default value**

""

Transform downstream attributes of table walk and queue transactions.

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**HWATTR\_KIND\_0**

PBHA information

**kind**

Transaction kind

- For a read:

**0/1**

L1STE/STE

**2/3**

L1CD/CD

**4/5**

S1/S2 TTD (including CAS)

**6**

CMDQ

**7**

VMS

**11/12**

LOGPT/L1GPT

**13/14**

LODPT/L1DPT

- For a write:

**0**

EVENTQ

**1**

PRIQ

ExtendedID/ManagerID64/MasterID/UserFlags start with values {0, 0xFFFFFFFF, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.

**Note**

See also `output_attribute_transform` and `msi_attribute_transform`.

**version****Type**

string

**Default value**

"r0p0"

The version of this product.

Valid values are:

- r0p0
- r1p0

**wait\_cmdq\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing something on the command queue. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffff)) - 1]$ .

**wait\_eventq\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing something on the event queue. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffff)) - 1]$ .

**wait\_misc\_async\_actions\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing an async action that could be delayed is performed. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffff)) - 1]$ .

**wait\_msi\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before sending an MSI. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffff)) - 1]$ .

**wait\_pri\_req\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before processing a PRI Request. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffff)) - 1]$ .

**wait\_pri\_resp\_ticks****Type**

uint64\_t

**Default value**

1

This is the time to wait before sending a PRI Response back to the PCIe subsystem. When a PRI Response is an auto-response then the ATC might immediately make a new ATS request, that immediately fails, immediately makes a PRI Request, or auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

### 3.10.64 MMU\_720AE

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1101: IP revisions support**

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Limitations**

- No power control
- AMBA stash operations, destructive read and destructive hint operations are not supported on PVBUS and also are not supported by the device.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMC_CBEID0` fields. The PMU is intended for demonstration purposes only and for driver development.
- Limited RAS support
- The SYSCO interface is not implemented
- The low power interface is not implemented
- The IMP DEF MPAM register file is not implemented. This controls how the internal resources of the MMU-720AE are partitioned.
- `tcu_sid[31:0]` is not modelled, instead the parameter `smmu_msi_device_id` is used.
- The HWATTR side-band signal is available by configuring `output_attribute_transform`, `msi_attribute_transform` and `tw_qs_attribute_transform` with `HWATTR_KIND_0`. For some transactions HWATTR comes from the `SMMU_S_AGBPA[3:0]`. In the hardware, this register has an 'Update' bit[31] that should be written as 1 and will be turned to zero when all transactions



using the old value have completed. The model does not implement this behavior and the Update bit is **RAZ/WI**.

- Any configuration parameter listed in the TRM but not shown in this file is not supported.
- `TCU_CTRL_AUX0-55` registers are modeled but unlike the TRM, reset values for these registers are 0. There is no functionality associated with these registers.
- There is no functionality associated with `TCU_ROOT_CTRL` register field `DIS_DVM`. Note: If `DIS_DVM` is set to 1, an error message is thrown which can be disabled by setting `all_error_messages_through_trace` to true.
- The model does not support the `eventoack` signal. In integration mode, the value of the field `eventoack` (bit 0) in `ITIN_PIU` will match the value of the `evento` signal.
- There is no functionality associated with `TCU_NODE_STATUS` register field `ATSv`, which indicates whether the node implements DTI-ATSv4.

## Notes

- A bad configuration renders the model inactive.
- Some configurations can be adjusted by configuration pins. These are only sampled at the negative edge of the reset pin. If you want to use these pins then you must drive them before sending a negative edge on the reset pin. During simulation reset the component driving them must also drive this transition again.
- Debug reads to the registers do not disturb the state.
- Writes to registers with Update flags, including debug writes, are ignored if the Update flag is already set to one.
- Debug and real accesses to the registers must be 32 bits or 64 bits.
- The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping used is remembered by the bus infrastructure and is used for subsequent sufficiently similar transactions without requiring intervention from the SMMU model, and is not traced, for instance.

The range of addresses that the mapping is valid for is determined by the SMMU model, depending on architectural and model-implementation details. However, as it is unaware of any sign-extension or the mapping that the SoC does, the SoC is responsible for subsequently narrowing the range of addresses for which this mapping is valid. Typically, this is done automatically when using `PVBusMapper`.

- The hardware is a distributed SMMU and is divided into:
  - A single Translation Control Unit (TCU)
    - Has a port for the programming interface of the SMMU
    - Receives DVM messages
    - Does all the page walking, queue manipulation, etc.
  - One or more Translation Bus Units (TBUs)
    - Translate transactions from upstream (client) device into downstream transactions.
  - Zero or more connections to PCIe Root Complexes (PCIe-RCs)

- There can be a total of 62 TBUs and PCIe-RCs attached.
- An interconnect connecting the TBUs, PCIe-RCs to the TCU.
- A PCIe-RC has one connection to the TCU to make ATS requests but the PCIe-RC uses one or more TBUs to transform the transactions and pass them to the memory system. In the model, those TBUs are configured by the parameter `list_of_pcie_mode`. The SMMU does not know which TBUs a particular PCIe-RC is attached to.
- The TBU ORs a value into each StreamID that it receives. In the model, this is configured by the parameters:
  - `list_of_s_sid_high_at_bitpos0`
  - `list_of_ns_sid_high_at_bitpos0`
- The TCU, TBU, and the interconnect are all represented by this single model component.
- In the model, a pair of ports `tbs_pvbuss_s[i]/tbm_pvbuss_m[i]` represent a TBU 'i' or the `tbs_pvbuss_s[i]` represents an incoming connection for a PCIe-RC. The corresponding reverse connection from the TCU to the PCIe-RC is by a special bus called `pvbus_id_routed_m` that is used to transport ATC Invalidates to the PCIe-RC.
  - In order to reduce system construction complexity the `tbs_pvbuss_s[i]/tbm_pvbuss_m[i]` also acts as a TBU so that the PCIe-RC need not separate its normal transactions and its ATS requests.
  - However, ATC Invalidates are only sent to a port which appears in `list_of_pcie_rc`. It should be uniquely decoded to a single port based on `list_of_ns_sid_high_at_bitpos0` and those ATC Invalidates must be routed to the correct PCIe subsystem in order to invalidate the cache of ATS Response in the subsystem. Thus all TBUs that a PCIe-RC uses must have a unique reverse mapping from stream id to port.
- The pin `sup_oas` is not supported, instead it is a parameter as it is assumed that it would be tied to a fixed value in any specific platform.
- The hardware only has a single cacheability attribute for input transactions, but PVBUS transports both inner and outer cacheability.
- For non-PCIe-mode TBUs (i.e. their index does not appear in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - For non-cache maintenance operations:
    - If the input attribute is any type of device then it is well defined as being outer-shared and Device-nGnRnE or Device-nGnRE. There is no support for Gathering or Reordering.
    - If the outer cacheable input attribute is normal then if it is Write-back then this is converted to Inner Write-back Outer Write-back (iWB-oWB) with the desired shareability. No Transient hint is supported and is always treated as non-transient.
    - This leaves all other normal memory types that are mapped to Inner Normal Non-cacheable, Outer Normal Non-cacheable outer shared (iNC-oNC-osh).
- Thus the upstream devices must present the cacheability in the *outer* cacheability attribute on PVBUS if it is cacheable. If it is a device type then both the inner and outer attributes must be set to the same. If it is iNC-oNC-osh then it must be presented as such.
- For PCIe-mode TBUs (i.e. their index appears in `list_of_pcie_mode` or `list_of_pcie_rc`):

- Input transactions are from PCIe and the only indication of the memory type is in the NoSnoop bit of the transaction. No shareability is transported.
  - NoSnoop interpreted as iNC-oNC-osh
  - ! NoSnoop interpreted as iWB-oWB-ish (note inner shareable)
- If a NoSnoop transaction has an attribute transform applied to it and the result of the transform is weaker than iNC-oNC-osh then it is forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it is forced to iNC-oNC-osh. However, if the page tables transformed it to a device type then, as all device types are stronger than iNC-oNC-osh then it exits the SMMU as the device type.
- In the model, transactions are classified by their incoming memory attributes as to whether they are NoSnoop or not and then normalized appropriately:
  - iWB-oWB-any-shareability transactions are interpreted as ! NoSnoop and therefore are normalized to iWB-oWB-ish.
  - Anything else is considered NoSnoop and therefore is normalized to iNC-oNC-osh.
- Translated accesses also suffer the same interpretation to determine NoSnoop and how they are normalized. Thus they could enter the system with attributes different to if they were Untranslated Accesses translated by normal means.
- It is expected that there is a component downstream of the SMMU that is aware of the system address map and will override the memory type to 'device' for any transaction that accesses a peripheral.
- The hardware has a single cacheability on input and, for transactions that are neither cache-maintenance operations nor PCIe transactions, normalizes the input to an architectural form before performing the SMMUv3 architectural transform:
  - Any device type is left untouched (the input can only represent Device-nGnRE and Device-nGnRE).
  - If the input is Write-back (WB) then it is normalized to iWB-oWB with the incoming shareability.
  - If the input is anything else it is normalized to iNC-oNC-osh.
- The model accepts full architectural attributes of two levels of cacheability and so has to decide how to interpret this in terms of the hardware. For transactions that are not cache maintenance operations, the model replicates the outer attribute into the inner attribute and then performs the normalization that the hardware does.
- The hardware normalizes the architectural output attributes and outputs a single level of cacheability and a user flag (OC) specifying if the architectural attributes were cacheable in the outer cacheable domain. If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. If the original transaction was NoSnoop, then any weaker memory type is strengthened to iNC-oNC-osh so apply the following transform:

```

if      iWB-oWB-nsh/ish/osh      then output WB-nsh/ish/osh, OC = 1
else if i(NC/WT/WB)-o(WB/WT)    then output NC-Sys,      OC = 1
else if i(NC/WT/WB)-oNC         then output NC-Sys,      OC = 0
else if Device-(GRE/nGRE/nGnRE) then output DV-Sys,      OC = 0
else                               output SO-Sys,      OC = 0

```

- The model only normalizes according to PCIe but otherwise leaves the architectural attributes intact on the output bus.
- MSIs are issued on the `qtw_pvbus_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `MasterID=0xFFFFffff`.

In the hardware, there is no way of distinguishing Event Queue writes from MSI writes, however, this provides a mechanism that if the model system needs to distinguish then it can. MPAM and MEC attributes are provided by the parameters:

- `mpam_attribute_transform`
- `mec_attribute_transform` (not all versions support MEC)
- The model supports architectural features and registers matching `rOp0-00eac0`.

### Security State Determination (SSD)

The model uses the term SSD to mean the security state that the transaction, register, structure, or event belongs to. In the SMMUv3 architecture, the sideband signal `SEC_SID` holds this information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one-bit signal. With RME, in the hardware, it was expanded to two bits. To retain backwards compatibility in the model, `SEC_SID` remains as one-bit in the parameter `howto_identify`, but the second bit can be expressed with `SEC_SID_bit_1`, and its negative logic version `nSEC_SID_bit_1`.

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

### Parameters for parsing transaction attributes

Some of the parameters are strings that share a common parsing format for extracting from and placing information into the transaction's attributes.

They can extract and place information into the following fields of a transaction's attributes:

- `ManagerID64` (64 bits)
- `MasterID` (32 bits) (This is the lower 32 bits of `ManagerID64`)
- `ExtendedID` (64 bits)
- `UserFlags` (32 bits)

The string parameter is parsed as a comma-separated list of:

```
lhs_expression=rhs_expression
```

The `lhs_expression` and `rhs_expression` can be an entire symbol, for example:

```
StreamID
```

or a bit, or bit slice:

```
StreamID[16]
StreamID[31:20]
```

In `rhs_expression`, numeric constants (or slices of numeric constants) are also allowed:

```
StreamID[16]=1
StreamID[16]=10[2]
```

A single symbol might be assigned from multiple non-contiguous arrays of bits from a mix of different RHS symbols:

```
StreamID[16]=1, StreamID[31:30]=ExtendedID[1:0],
StreamID[15:0]=UserFlags[31:16], ...
```

In those cases where a left hand symbol can also appear on the right hand side, it is possible to swap bits and transform the symbol. For example, the following expression swaps bits 0 and 1 of the `ExtendedID`:

```
ExtendedID[0]=ExtendedID[1], ExtendedID[1]=ExtendedID[0]
```

Any bits of the attributes that have no transform specified are retained from the input.

The `lhs_expression` and `rhs_expression` must have the same bit width.

## Iris and MTI instances for MMU\_720AE

This model has the following Iris instances:

**Table 3-1103: MMU\_720AE Iris instances**

InstanceName	ComponentName
MMU_720AE	MMU_720AE
MMU_720AE.register_file[0]	PVBusSlave
MMU_720AE.service_request_tbu[0]	PVBusSlave
MMU_720AE.service_request_tbu[10]	PVBusSlave
MMU_720AE.service_request_tbu[11]	PVBusSlave
MMU_720AE.service_request_tbu[12]	PVBusSlave
MMU_720AE.service_request_tbu[13]	PVBusSlave
MMU_720AE.service_request_tbu[14]	PVBusSlave
MMU_720AE.service_request_tbu[15]	PVBusSlave

InstanceName	ComponentName
MMU_720AE.service_request_tbu[16]	PVBusSlave
MMU_720AE.service_request_tbu[17]	PVBusSlave
MMU_720AE.service_request_tbu[18]	PVBusSlave
MMU_720AE.service_request_tbu[19]	PVBusSlave
MMU_720AE.service_request_tbu[1]	PVBusSlave
MMU_720AE.service_request_tbu[20]	PVBusSlave
MMU_720AE.service_request_tbu[21]	PVBusSlave
MMU_720AE.service_request_tbu[22]	PVBusSlave
MMU_720AE.service_request_tbu[23]	PVBusSlave
MMU_720AE.service_request_tbu[24]	PVBusSlave
MMU_720AE.service_request_tbu[25]	PVBusSlave
MMU_720AE.service_request_tbu[26]	PVBusSlave
MMU_720AE.service_request_tbu[27]	PVBusSlave
MMU_720AE.service_request_tbu[28]	PVBusSlave
MMU_720AE.service_request_tbu[29]	PVBusSlave
MMU_720AE.service_request_tbu[2]	PVBusSlave
MMU_720AE.service_request_tbu[30]	PVBusSlave
MMU_720AE.service_request_tbu[31]	PVBusSlave
MMU_720AE.service_request_tbu[32]	PVBusSlave
MMU_720AE.service_request_tbu[33]	PVBusSlave
MMU_720AE.service_request_tbu[34]	PVBusSlave
MMU_720AE.service_request_tbu[35]	PVBusSlave
MMU_720AE.service_request_tbu[36]	PVBusSlave
MMU_720AE.service_request_tbu[37]	PVBusSlave
MMU_720AE.service_request_tbu[38]	PVBusSlave
MMU_720AE.service_request_tbu[39]	PVBusSlave
MMU_720AE.service_request_tbu[3]	PVBusSlave
MMU_720AE.service_request_tbu[40]	PVBusSlave
MMU_720AE.service_request_tbu[41]	PVBusSlave
MMU_720AE.service_request_tbu[42]	PVBusSlave
MMU_720AE.service_request_tbu[43]	PVBusSlave
MMU_720AE.service_request_tbu[44]	PVBusSlave
MMU_720AE.service_request_tbu[45]	PVBusSlave
MMU_720AE.service_request_tbu[46]	PVBusSlave
MMU_720AE.service_request_tbu[47]	PVBusSlave
MMU_720AE.service_request_tbu[48]	PVBusSlave
MMU_720AE.service_request_tbu[49]	PVBusSlave
MMU_720AE.service_request_tbu[4]	PVBusSlave
MMU_720AE.service_request_tbu[50]	PVBusSlave

InstanceName	ComponentName
MMU_720AE.service_request_tbu[51]	PVBusSlave
MMU_720AE.service_request_tbu[52]	PVBusSlave
MMU_720AE.service_request_tbu[53]	PVBusSlave
MMU_720AE.service_request_tbu[54]	PVBusSlave
MMU_720AE.service_request_tbu[55]	PVBusSlave
MMU_720AE.service_request_tbu[56]	PVBusSlave
MMU_720AE.service_request_tbu[57]	PVBusSlave
MMU_720AE.service_request_tbu[58]	PVBusSlave
MMU_720AE.service_request_tbu[59]	PVBusSlave
MMU_720AE.service_request_tbu[5]	PVBusSlave
MMU_720AE.service_request_tbu[60]	PVBusSlave
MMU_720AE.service_request_tbu[61]	PVBusSlave
MMU_720AE.service_request_tbu[62]	PVBusSlave
MMU_720AE.service_request_tbu[63]	PVBusSlave
MMU_720AE.service_request_tbu[6]	PVBusSlave
MMU_720AE.service_request_tbu[7]	PVBusSlave
MMU_720AE.service_request_tbu[8]	PVBusSlave
MMU_720AE.service_request_tbu[9]	PVBusSlave
MMU_720AE.tbu[0]	PVBusMapper
MMU_720AE.tbu[10]	PVBusMapper
MMU_720AE.tbu[11]	PVBusMapper
MMU_720AE.tbu[12]	PVBusMapper
MMU_720AE.tbu[13]	PVBusMapper
MMU_720AE.tbu[14]	PVBusMapper
MMU_720AE.tbu[15]	PVBusMapper
MMU_720AE.tbu[16]	PVBusMapper
MMU_720AE.tbu[17]	PVBusMapper
MMU_720AE.tbu[18]	PVBusMapper
MMU_720AE.tbu[19]	PVBusMapper
MMU_720AE.tbu[1]	PVBusMapper
MMU_720AE.tbu[20]	PVBusMapper
MMU_720AE.tbu[21]	PVBusMapper
MMU_720AE.tbu[22]	PVBusMapper
MMU_720AE.tbu[23]	PVBusMapper
MMU_720AE.tbu[24]	PVBusMapper
MMU_720AE.tbu[25]	PVBusMapper
MMU_720AE.tbu[26]	PVBusMapper
MMU_720AE.tbu[27]	PVBusMapper
MMU_720AE.tbu[28]	PVBusMapper

InstanceName	ComponentName
MMU_720AE.tbu[29]	PVBusMapper
MMU_720AE.tbu[2]	PVBusMapper
MMU_720AE.tbu[30]	PVBusMapper
MMU_720AE.tbu[31]	PVBusMapper
MMU_720AE.tbu[32]	PVBusMapper
MMU_720AE.tbu[33]	PVBusMapper
MMU_720AE.tbu[34]	PVBusMapper
MMU_720AE.tbu[35]	PVBusMapper
MMU_720AE.tbu[36]	PVBusMapper
MMU_720AE.tbu[37]	PVBusMapper
MMU_720AE.tbu[38]	PVBusMapper
MMU_720AE.tbu[39]	PVBusMapper
MMU_720AE.tbu[3]	PVBusMapper
MMU_720AE.tbu[40]	PVBusMapper
MMU_720AE.tbu[41]	PVBusMapper
MMU_720AE.tbu[42]	PVBusMapper
MMU_720AE.tbu[43]	PVBusMapper
MMU_720AE.tbu[44]	PVBusMapper
MMU_720AE.tbu[45]	PVBusMapper
MMU_720AE.tbu[46]	PVBusMapper
MMU_720AE.tbu[47]	PVBusMapper
MMU_720AE.tbu[48]	PVBusMapper
MMU_720AE.tbu[49]	PVBusMapper
MMU_720AE.tbu[4]	PVBusMapper
MMU_720AE.tbu[50]	PVBusMapper
MMU_720AE.tbu[51]	PVBusMapper
MMU_720AE.tbu[52]	PVBusMapper
MMU_720AE.tbu[53]	PVBusMapper
MMU_720AE.tbu[54]	PVBusMapper
MMU_720AE.tbu[55]	PVBusMapper
MMU_720AE.tbu[56]	PVBusMapper
MMU_720AE.tbu[57]	PVBusMapper
MMU_720AE.tbu[58]	PVBusMapper
MMU_720AE.tbu[59]	PVBusMapper
MMU_720AE.tbu[5]	PVBusMapper
MMU_720AE.tbu[60]	PVBusMapper
MMU_720AE.tbu[61]	PVBusMapper
MMU_720AE.tbu[62]	PVBusMapper
MMU_720AE.tbu[63]	PVBusMapper



InstanceName	ComponentName
MMU_720AE.tbv[6]	PVBusMapper
MMU_720AE.tbv[7]	PVBusMapper
MMU_720AE.tbv[8]	PVBusMapper
MMU_720AE.tbv[9]	PVBusMapper

This model has the following MTI trace components:

**Table 3-1104: MMU\_720AE MTI instances**

InstanceName	ComponentName
MMU_720AE	MMU_720AE
MMU_720AE.register_file[0]	PVBusSlave
MMU_720AE.service_request_tbv[0]	PVBusSlave
MMU_720AE.service_request_tbv[10]	PVBusSlave
MMU_720AE.service_request_tbv[11]	PVBusSlave
MMU_720AE.service_request_tbv[12]	PVBusSlave
MMU_720AE.service_request_tbv[13]	PVBusSlave
MMU_720AE.service_request_tbv[14]	PVBusSlave
MMU_720AE.service_request_tbv[15]	PVBusSlave
MMU_720AE.service_request_tbv[16]	PVBusSlave
MMU_720AE.service_request_tbv[17]	PVBusSlave
MMU_720AE.service_request_tbv[18]	PVBusSlave
MMU_720AE.service_request_tbv[19]	PVBusSlave
MMU_720AE.service_request_tbv[1]	PVBusSlave
MMU_720AE.service_request_tbv[20]	PVBusSlave
MMU_720AE.service_request_tbv[21]	PVBusSlave
MMU_720AE.service_request_tbv[22]	PVBusSlave
MMU_720AE.service_request_tbv[23]	PVBusSlave
MMU_720AE.service_request_tbv[24]	PVBusSlave
MMU_720AE.service_request_tbv[25]	PVBusSlave
MMU_720AE.service_request_tbv[26]	PVBusSlave
MMU_720AE.service_request_tbv[27]	PVBusSlave
MMU_720AE.service_request_tbv[28]	PVBusSlave
MMU_720AE.service_request_tbv[29]	PVBusSlave
MMU_720AE.service_request_tbv[2]	PVBusSlave
MMU_720AE.service_request_tbv[30]	PVBusSlave
MMU_720AE.service_request_tbv[31]	PVBusSlave
MMU_720AE.service_request_tbv[32]	PVBusSlave
MMU_720AE.service_request_tbv[33]	PVBusSlave
MMU_720AE.service_request_tbv[34]	PVBusSlave
MMU_720AE.service_request_tbv[35]	PVBusSlave

InstanceName	ComponentName
MMU_720AE.service_request_tbu[36]	PVBusSlave
MMU_720AE.service_request_tbu[37]	PVBusSlave
MMU_720AE.service_request_tbu[38]	PVBusSlave
MMU_720AE.service_request_tbu[39]	PVBusSlave
MMU_720AE.service_request_tbu[3]	PVBusSlave
MMU_720AE.service_request_tbu[40]	PVBusSlave
MMU_720AE.service_request_tbu[41]	PVBusSlave
MMU_720AE.service_request_tbu[42]	PVBusSlave
MMU_720AE.service_request_tbu[43]	PVBusSlave
MMU_720AE.service_request_tbu[44]	PVBusSlave
MMU_720AE.service_request_tbu[45]	PVBusSlave
MMU_720AE.service_request_tbu[46]	PVBusSlave
MMU_720AE.service_request_tbu[47]	PVBusSlave
MMU_720AE.service_request_tbu[48]	PVBusSlave
MMU_720AE.service_request_tbu[49]	PVBusSlave
MMU_720AE.service_request_tbu[4]	PVBusSlave
MMU_720AE.service_request_tbu[50]	PVBusSlave
MMU_720AE.service_request_tbu[51]	PVBusSlave
MMU_720AE.service_request_tbu[52]	PVBusSlave
MMU_720AE.service_request_tbu[53]	PVBusSlave
MMU_720AE.service_request_tbu[54]	PVBusSlave
MMU_720AE.service_request_tbu[55]	PVBusSlave
MMU_720AE.service_request_tbu[56]	PVBusSlave
MMU_720AE.service_request_tbu[57]	PVBusSlave
MMU_720AE.service_request_tbu[58]	PVBusSlave
MMU_720AE.service_request_tbu[59]	PVBusSlave
MMU_720AE.service_request_tbu[5]	PVBusSlave
MMU_720AE.service_request_tbu[60]	PVBusSlave
MMU_720AE.service_request_tbu[61]	PVBusSlave
MMU_720AE.service_request_tbu[62]	PVBusSlave
MMU_720AE.service_request_tbu[63]	PVBusSlave
MMU_720AE.service_request_tbu[6]	PVBusSlave
MMU_720AE.service_request_tbu[7]	PVBusSlave
MMU_720AE.service_request_tbu[8]	PVBusSlave
MMU_720AE.service_request_tbu[9]	PVBusSlave
MMU_720AE.tb[0]	PVBusMapper
MMU_720AE.tb[10]	PVBusMapper
MMU_720AE.tb[11]	PVBusMapper
MMU_720AE.tb[12]	PVBusMapper

InstanceName	ComponentName
MMU_720AE.tbu[13]	PVBusMapper
MMU_720AE.tbu[14]	PVBusMapper
MMU_720AE.tbu[15]	PVBusMapper
MMU_720AE.tbu[16]	PVBusMapper
MMU_720AE.tbu[17]	PVBusMapper
MMU_720AE.tbu[18]	PVBusMapper
MMU_720AE.tbu[19]	PVBusMapper
MMU_720AE.tbu[1]	PVBusMapper
MMU_720AE.tbu[20]	PVBusMapper
MMU_720AE.tbu[21]	PVBusMapper
MMU_720AE.tbu[22]	PVBusMapper
MMU_720AE.tbu[23]	PVBusMapper
MMU_720AE.tbu[24]	PVBusMapper
MMU_720AE.tbu[25]	PVBusMapper
MMU_720AE.tbu[26]	PVBusMapper
MMU_720AE.tbu[27]	PVBusMapper
MMU_720AE.tbu[28]	PVBusMapper
MMU_720AE.tbu[29]	PVBusMapper
MMU_720AE.tbu[2]	PVBusMapper
MMU_720AE.tbu[30]	PVBusMapper
MMU_720AE.tbu[31]	PVBusMapper
MMU_720AE.tbu[32]	PVBusMapper
MMU_720AE.tbu[33]	PVBusMapper
MMU_720AE.tbu[34]	PVBusMapper
MMU_720AE.tbu[35]	PVBusMapper
MMU_720AE.tbu[36]	PVBusMapper
MMU_720AE.tbu[37]	PVBusMapper
MMU_720AE.tbu[38]	PVBusMapper
MMU_720AE.tbu[39]	PVBusMapper
MMU_720AE.tbu[3]	PVBusMapper
MMU_720AE.tbu[40]	PVBusMapper
MMU_720AE.tbu[41]	PVBusMapper
MMU_720AE.tbu[42]	PVBusMapper
MMU_720AE.tbu[43]	PVBusMapper
MMU_720AE.tbu[44]	PVBusMapper
MMU_720AE.tbu[45]	PVBusMapper
MMU_720AE.tbu[46]	PVBusMapper
MMU_720AE.tbu[47]	PVBusMapper
MMU_720AE.tbu[48]	PVBusMapper

InstanceName	ComponentName
MMU_720AE.tbv[49]	PVBusMapper
MMU_720AE.tbv[4]	PVBusMapper
MMU_720AE.tbv[50]	PVBusMapper
MMU_720AE.tbv[51]	PVBusMapper
MMU_720AE.tbv[52]	PVBusMapper
MMU_720AE.tbv[53]	PVBusMapper
MMU_720AE.tbv[54]	PVBusMapper
MMU_720AE.tbv[55]	PVBusMapper
MMU_720AE.tbv[56]	PVBusMapper
MMU_720AE.tbv[57]	PVBusMapper
MMU_720AE.tbv[58]	PVBusMapper
MMU_720AE.tbv[59]	PVBusMapper
MMU_720AE.tbv[5]	PVBusMapper
MMU_720AE.tbv[60]	PVBusMapper
MMU_720AE.tbv[61]	PVBusMapper
MMU_720AE.tbv[62]	PVBusMapper
MMU_720AE.tbv[63]	PVBusMapper
MMU_720AE.tbv[6]	PVBusMapper
MMU_720AE.tbv[7]	PVBusMapper
MMU_720AE.tbv[8]	PVBusMapper
MMU_720AE.tbv[9]	PVBusMapper

## Ports for MMU\_720AE

**Table 3-1105: Ports**

Name	Protocol	Type	Description
axi_stream_msi_addr_to_match_s	Value_64	Slave	Any SMMU-originated MSI which exactly matches the address in this port will be sent through the AXI stream port axi_stream_msi_m which is usually connected to the GIC through axi_stream_msi_s. As MSIs are 32 bit aligned then if bits[1:0] != 0 or the address is above OAS then this is effectively disabled. NOTE that the model does not support tcu_sid[31:0] which is the MSI DeviceID to send on axi_stream_msi_m. Instead the parameter smmu_msi_device_id is used. See also the parameters: axi_stream_msi_TID and axi_stream_msi_TDEST. The default value of this port is set by the parameter axi_stream_msi_addr_to_match. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
axi_stream_msi_m	PVBus	Master	Manager port used for sending SMMU originated MSIs directly to the GIC
clk_in	ClockSignal	Slave	Clock signal (in RTL aclk) This is a clock time-base used by the TCU to spread some of its processing over time, if enabled by the wait_* parameters. The clock must always be connected.

Name	Protocol	Type	Description
cmd_sync_irpt_ns	Signal	Master	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of SIG_IRQ.
cmd_sync_irpt_r	Signal	Master	Pulsed interrupt output signal for realm CMD_SYNC having a completion signal of SIG_IRQ. This pin exists in r0 but is tied off to 0. It is implemented in r1.
cmd_sync_irpt_s	Signal	Master	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
event_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
event_q_irpt_r	Signal	Master	Pulsed interrupt output signal for the realm event queue becoming non-empty. This pin exists in r0 but is tied off to 0. It is implemented in r1.
event_q_irpt_s	Signal	Master	Pulsed interrupt output signal for the secure event queue becoming non-empty.
evento	Signal	Master	Event signal This aligns with the eventoreq signal on the RTL. The eventoack signal is not supported.
global_irpt_ns	Signal	Master	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signalling an error.
global_irpt_r	Signal	Master	Pulsed interrupt output signal for realm SMMU_R_GERROR(N) signalling an error. This pin exists in r0 but is tied off to 0. It is implemented in r1.
global_irpt_s	Signal	Master	Pulsed interrupt output signal for secure SMMU_S_GERROR(N) signalling an error.
gpf_far	Signal	Master	An error becomes active in SMMU_ROOT_GPF_FAR.
gpt_cfg_far	Signal	Master	An error becomes active in SMMU_ROOT_GPT_CFG_FAR.
identify	SMMUV3AEMIdentifyProtocol	Master	Map the transaction to the tuple (StreamID, SubStreamID, SubStreamIDValid, SSD) The StreamID that is produced by the implementation of this protocol is not the final StreamID. The final StreamID is produced by using the list_of_ns_sid_high_at_bitpos0/list_of_s_sid_high_at_bitpos0 parameter to map the StreamID based on the upstream port index. Also see the parameter howto_identify which can replace the functionality of this port under certain circumstances.
logptsz_s	Value	Slave	RME: This is a four bit signal that encodes the region size that a single LOGPT entry covers. The default value of this port is derived from the parameter rme_logpt_entry_covers_log2size_in_bytes which is in a different format to the port. If a valid value is driven then it will be put in the field SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ. The port uses the same encoding as the field. If an invalid value is driven to this port and legacy_tz_en is low then the value is reported in the SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ and then all transactions will fault with a GPT Configuration fault (gpt_cfg_far). This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.

Name	Protocol	Type	Description
legacy_tz_en	Signal	Slave	Tie this high to get non-RME behaviour. On the real hardware, then each of the TCUs and the TBUs have a legacy_tz_en and they must all be driven to the same value. In the model, we only have a single version of this pin. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
ns_gbpa_abort_init	Signal	Slave	This port is an Non-secure global bypass. The ns_gbpa_abort_init signal sets the reset value of SMMU_GBPA.ABORT: 0 - On reset, do not abort all incoming transactions. 1 - On reset, abort all incoming transactions. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
pri_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the non-secure PRI queue becoming non-empty.
pri_q_irpt_r	Signal	Master	Pulsed interrupt output signal for the realm PRI queue becoming non-empty. This pin exists in r0 but is tied off to 0. It is implemented in r1.
prog_pvbus_s	PVBus	Slave	Register subordinate port (in RTL PROG)
pvbus_id_routed_m[62]	PVBus	Master	This is a special "id-routed" port for transmitting ATC invalidates upstream into the PCIe EndPoints, it is not a normal bus. See parameter output_id_routed_transform. The FastModels ATC Invalidate and PRI Response protocol specifies how to route and deal with this port.
qtw_pvbus_m	PVBus	Master	Manager port used for Table Walks, MSIs and Queue access. Note that although this is a manager port, it can still receive DVM messages.
s_gbpa_abort_init	Signal	Slave	This port is an secure global bypass. The s_gbpa_abort_init signal sets the reset value of SMMU_S_GBPA.ABORT: 0 - On reset, do not abort all incoming transactions. 1 - On reset, abort all incoming transactions. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sec_override	Signal	Slave	Allow certain registers to be accessible to non-secure accesses from reset, as described in the TCU_SCR register. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_btm	Signal	Slave	System supports BTM and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If BTM (Broadcast Table Maintenance) is not supported then DVM messages will be ignored. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_cohacc	Signal	Slave	System supports COHACC and will be reflected in the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_httu	Signal	Slave	System supports HTTU and will be reflected in the value of SMMU_IDR0.HTTU: - 0b00 (HTTU not supported) if sup_httu is 0 - 0b10 (update of AF and Dirty flags supported) if sup_httu is 1 The default value for this pin is 1. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.

Name	Protocol	Type	Description
sup_sev	Signal	Slave	System supports SEV and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
tbm_pvbus_m[62]	PVBus	Master	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered tbs_pvbus_s[] port.
tbs_pvbus_s[62]	PVBus	Slave	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
tbu_crit_err[62]	Signal	Master	Critical error. This cannot occur in the model except by using an Integration Register to generate it.
tbu_fmurstdisable[62]	Signal	Slave	TBU Preserve FMU error record The TBUs have independent signals to enable a Warm Reset which preserves the state of the FMU node registers. Warm reset is enabled via the logical AND of the tbu_fmurstdisable signal and the rising edge of the tbu_reset_in signal. This signal must be asserted and stable prior to reset to enable a warm reset.
tbu_pmu_irpt[62]	Signal	Master	TBU Performance Monitoring Unit interrupt, one per TBU.
tbu_pmusnapshot_ack[62]	Signal	Master	PMU snapshot interface for the TBU, ack a snapshot.
tbu_pmusnapshot_req[62]	Signal	Slave	PMU snapshot interface for the TBU, request a snapshot.
tbu_ras_cri[62]	Signal	Master	Critical error interrupt for RAS events from the TBU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE that in the MMU-720AE model this is called tbu_ras_et_cri.
tbu_ras_eri[62]	Signal	Master	Error Recovery Interrupt for RAS events from the TBU. This is used for uncorrected errors. NOTE that in the TRM of MMU-720AE model this is called tbu_ras_et_eri.
tbu_ras_fhi[62]	Signal	Master	Fault Handling Interrupt for RAS events from the TBU. Usually this is for corrected errors. NOTE that in the TRM of MMU-720AE model this is called tbu_ras_et_fhi.
tbu_ras_lt_cri[62]	Signal	Master	Level triggered critical error interrupt for RAS events from the TBU.
tbu_ras_lt_eri[62]	Signal	Master	Level triggered error recovery interrupt for RAS events from the TBU.
tbu_ras_lt_fhi[62]	Signal	Master	Level triggered fault handling interrupt for RAS events from the TBU.
tbu_ras_lt_irpt_v[62]	Signal	Master	Level triggered valid output for connection to System RAS agents. Asserted when the RAS record contains at least one valid error.

Name	Protocol	Type	Description
tbu_reset_in[62]	Signal	Slave	Reset signals The TBUs can have independent reset signals. However, TCU reset will result in the reset of all TBUs. This behavior is unlike SMMU RTL implementations which do allow for independent reset of the TCU separate from TBUs. Each signal tbu_reset_in[n] corresponds to the TBU using tbs_pvbus_s[n]/tbs_pvbus_s[n] pair. If the SMMU receives a transaction whilst the TBU is expected to be in reset then it will complain using the ArchMsg.Warning.warning trace source. Those tbu_reset_in that correspond to a PCIe-RC connection can be connected to monitor the PCIe-RC's reset signal. If it receives an ATS request when in reset then it will complain in a similar way. You must connect these pins if you wish the TCU_NODE_STATUS for the nodes to be accurate (including any connected to the PCIe-RC).
tcu_fmurstdisable	Signal	Slave	TCU Preserve FMU error record The TCU has an independent signal to enable a Warm Reset which preserves the state of the FMU node registers. Warm reset is enabled via the logical AND of the tcu_fmurstdisable signal and the rising edge of the tcu_reset_in signal. This signal must be asserted and stable prior to reset to enable a warm reset.
tcu_pmu_irpt	Signal	Master	TCU Performance Monitoring Unit interrupt
tcu_pmusnapshot_ack	Signal	Master	PMU snapshot interface for the TCU, ack a snapshot.
tcu_pmusnapshot_req	Signal	Slave	PMU snapshot interface This is a four-phase handshake to have the corresponding PMCG perform a capture of the current counter values. PMU snapshot interface for the TCU, request a snapshot.
tcu_ras_cri	Signal	Master	Critical error interrupt for RAS events from the TCU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE that in the TRM of MMU-720AE model this is called tcu_ras_et_cri.
tcu_ras_eri	Signal	Master	Error Recovery Interrupt for RAS events from the TCU. This is used for uncorrected errors. NOTE that in the TRM of MMU-720AE model this is called tcu_ras_et_eri.
tcu_ras_fhi	Signal	Master	Fault Handling Interrupt for RAS events from the TCU. Usually this is for corrected errors. NOTE that in the TRM of MMU-720AE model this is called tcu_ras_et_fhi.
tcu_ras_lt_cri	Signal	Master	Level triggered critical error interrupt for RAS events from the TCU.
tcu_ras_lt_eri	Signal	Master	Level triggered error recovery interrupt for RAS events from the TCU.
tcu_ras_lt_fhi	Signal	Master	Level triggered fault handling interrupt for RAS events from the TCU.
tcu_ras_lt_irpt_v	Signal	Master	Level triggered valid output for connection to System RAS agents. Asserted when the RAS record contains at least one valid error.
tcu_reset_in	Signal	Slave	The reset signal to the TCU interface.



## Parameters for MMU\_720AE

### **TCUCFG\_DPT\_SUPPORT**

#### **Type**

bool

#### **Default value**

true

MMU-S3 r1 Parameter Only: Enable Device Permission Table (DPT), a mechanism to enforce the association between granules of physical address space and the memory footprint of virtual machines.

#### **0**

Realm DPT is disabled

#### **1**

Realm DPT is enabled (default)

### **TCUCFG\_DVM\_VAS**

#### **Type**

uint32\_t

#### **Default value**

53

Virtual address size used by the system. Once set, this value is discoverable using `TCU_SYSDISC35.TCUCFG_DVM_VAS`.

In hardware, it is important to get this parameter correct as it determines the DVM message format. If this doesn't match the PEs, DVM messages are misinterpreted and any TLBI operations performed are incorrectly applied.

The model uses a representation of DVM that does not depend on the VA size and so misconfiguring this has no effect other than on the system discovery register value. Accepted Values: 49 53

### **TCUCFG\_MECID\_WIDTH**

#### **Type**

uint32\_t

#### **Default value**

16

Memory Encryption Context (MEC) is a feature introduced in MMU-S3. The MECID is a 1-16 bit identifier that, if implemented, supports Memory Encryption Contexts for the Realm programming interface. The given value indicates the number of bits in the MECID. A value of 0 will disable MEC. Accepted Values: 0 4 8 12 16

**TCUCFG\_PARTID\_WIDTH****Type**

unsigned

**Default value**

9

The width of the MPAM PARTID on the bus.

The value 10 is just for MMU\_S3 r1.

See also parameter mpam\_attribute\_transform. Accepted Values: 1 6 9 10

**TCUCFG\_XLATE\_SLOTS****Type**

uint32\_t

**Default value**

512

Maximum number of outstanding stalled transactions that the SMMU supports.

**Note**

TCUCFG\_XLATE\_SLOTS must be  $\geq$  TCUCFG\_PTW\_SLOTS which is currently fixed to 512.

---

Accepted Values: 512 1024 2048 4096

**all\_error\_messages\_through\_trace****Type**

bool

**Default value**

false

Some conditions in the SMMU are so strange that the software programming the SMMU has done something wrong. At this point messages are output to either `ArchMsg.Error.*` or `ArchMsg.Warning.*` or to the error stream of the simulator. Outputting to the error stream of the simulator may cause it to return with a non-zero exit status.

If you set this option to true then instead of using the error stream of the simulator it will always use a trace stream allowing the simulation to exit with a zero exit status.

**axi\_stream\_msi\_TDEST****Type**

uint32\_t

**Default value**

0

ID of the AXI stream subordinate port on the GIC that receives SMMU originated MSIs sent directly. The name of the GIC port is `axi_stream_msi_s`.

**Note**

If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TID`

**`axi_stream_msi_TID`****Type**`uint32_t`**Default value**

0

ID of the AXI stream manager port that sends SMMU TCU originated MSIs directly to the GIC. The name of the SMMU port is `axi_stream_msi_m`.

**Note**

If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TDEST`

**`axi_stream_msi_addr_to_match`****Type**`uint64_t`**Default value**`0xFFFFffffffFFFFFF`

If the last two bits are 0, any SMMU-originated MSI which exactly matches the address will be sent through the AXI stream port `axi_stream_msi_m` which is usually connected to the GIC.

This parameter drives the value of the `axi_stream_msi_addr_to_match_s` port at simulation reset. For every reset after that, the value of the port will be sampled and used if changed.

**Note**

The entire address must match, including bits [1:0]. As MSIs are 32 bit aligned then if `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

See also:

- `axi_stream_msi_TID`
- `axi_stream_msi_TDEST`

### **behaviour\_of\_sampled\_at\_reset\_signals**

#### **Type**

unsigned

#### **Default value**

0

Some configuration signals into the SMMU are sampled on negedge of reset.

However, it can sometimes be hard to arrange to drive a configuration pin before the negedge of reset.

The configuration pins are sampled:

**0**

at negedge reset.

**1**

at negedge reset, but if a later change occurs at the same simulated time, and no transactions have occurred, then they will be resampled and the SMMU reset again.

### **cmdq\_max\_number\_of\_commands\_to\_buffer**

#### **Type**

uint32\_t

#### **Default value**

10

The command queues can buffer fetched commands before issuing them. This parameter is roughly the maximum number of commands to do this for. The programmer visible effects are that just because the CONS pointer shows a command has been consumed does not necessarily mean that it has been issued (and completed). Higher values accentuate this effect.

**enable\_device\_id\_checks****Type**

bool

**Default value**

true

If this parameter is true then the DeviceIDs seen by the GIC are:

- **for client devices**

$$\text{DeviceID} = \text{StreamID} + \text{translated\_device\_id\_base}$$

- **for SMMU-generated MSIs**

$$\text{smmu\_msi\_device\_id}$$

This parameter enables two checks:

- If the DeviceID is used in the `output_attribute_transform` parameter, if it overflows 32 bits then the model warns. If the DeviceID is not used, it is assumed that the external agent that forms the DeviceID warns if it overflows.
- If the SMMU supports MSIs, the model checks that the GIC is able to distinguish an MSI generated by the SMMU from one generated by a client device.

As the exact mechanism to determine the DeviceID is in the system and not necessarily under control of the SMMU then you can disable these warnings using this parameter.

See also the parameters: `output_attribute_transform` and `msi_attribute_transform`.

**hide\_warning\_ACCESSEN\_GPCEN\_set\_to\_1\_in\_a\_single\_write****Type**

bool

**Default value**

false

The architecture recommends against setting `SMMU_ROOT_CR0.ACCESSEN` and `SMMU_ROOT_CR0.GPCEN` to 1 in the same write if it is possible that a concurrent client device transaction could appear at the SMMU. This is because there could be a time window where the client device transaction sees effective values `ACCESSEN == 1` and `GPCEN == 0` and so would bypass GPC checking.

If your system cannot generate this possibility then you can set this parameter to true and turn off the warning that software has set both `ACCESSEN` and `GPCEN` to 1 at the same time.

**hide\_warning\_E0PD\_differs\_from\_what\_would\_be\_cached****Type**

bool

**Default value**

false

When this parameter is set to true, warnings that the effective EOPD value differs from what would be cached in the TLB are disabled. False (warnings are showed) by default.

### **hide\_warning\_NoStreamID\_transaction\_for\_unsupported\_PAS\_or\_MPAM\_SP**

#### **Type**

bool

#### **Default value**

false

When RME is not supported then a NoStreamID transaction with `PAS[1] == 1` or `MPAM_SP[1] == 1` is treated as though `PAS[1] == 0` and `MPAM_SP[1] == 0`. This is usually a system construction error and is not expected to occur.

The SMMU warns when this occurs, but the warning can be hidden by setting this parameter.

### **howto\_identify**

#### **Type**

string

#### **Default value**

"use-identify"

If `use-identify` then the SMMU uses the `identify` port to determine the `ssd`, `streamID`, `subStreamID`. Otherwise, this string extracts them from the transaction's attributes.

Examples:

```
SEC_SID=ExtendedID[63], SSV=ExtendedID[62], SubstreamID=ExtendedID[51:32],
StreamID=ExtendedID[31:0]
```

or

```
nSEC_SID=ExtendedID[63], StreamID=ExtendedID[55:24], nSSV=ExtendedID[20],
SubstreamID=ExtendedID[19:0]
StreamID[31:24]=0, StreamID[23:0]=ExtendedID[23:0], SSV=1[0], ...
```



#### **Note**

If you are not using the `use-identify` option then the configuration string is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

LHS Symbols:

#### **SIDV**

Indicates that the StreamID is valid.

**SSV**

Indicates that the SubstreamID is valid.

**SEC\_SID / SEC\_SID\_bit\_1**

Bits 0 and 1 respectively of the StreamID Security State

**StreamID**

(32 b) valid if `SIDV` is 1 or both `SIDV` and `nSIDV` are unused.

**SubstreamID**

(20 b) is valid if `ssv` is true.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

RHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming PVBUS transaction attributes

`SEC_SID` is 1b in the model. For RME systems, in hardware, then `SEC_SID` is 2b, in the model `SEC_SID_bit_1` represents bit[1].

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

Negative and positive logic symbols for the same attribute is an error.

The model uses the term SSD (Security State Determination) to mean which security state the transaction, register, structure, event, etc. belongs to.

In the SMMUV3 Architecture, the side-band signal `SEC_SID` holds the information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one bit signal. With RME, in the hardware, it was expanded to 2b. To retain backwards compatibility in the model, `SEC_SID` remains as 1b in this parameter, but the second bit can be expressed with `SEC_SID_bit_1` (and its negative logic version `nSEC_SID_bit_1`)

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

For those systems that support NoStreamID transactions, and `howto_identify` is not using the port, first the SMMU determines the value of `SIDV` or `nSIDV` to see if the transaction is a NoStreamID transaction (`SIDV == 0` or `nSIDV == 1`). If so then the rest of the `howto_identify` string is ignored as the information extracted relates only to StreamID transactions. Instead more information is extracted using the parameter `howto_identify_NoStreamID_extra_info`.

In AMBA systems, the equivalent signal to `SIDV` is called either `ARMMUVALID` or `AWMMUVALID`. Collectively they are known as `AxMMUVALID`.

### **howto\_identify\_NoStreamID\_extra\_info**

#### **Type**

string

#### **Default value**

""

The behavior of this parameter depends on `howto_identify`

- if it equals 'use-identify' then this must be "", otherwise there is an error.
- if it identifies a NoStreamID transaction (`SIDV=0`) then this parameter includes one or more of
  - `MPAM_SP`
  - `MPAM_PARTID`
  - `MPAM_PMG`
  - `MECID`
  - `HWATTR_KIND_0`
- in any other case, this parameter is ignored.

Fields set in this parameter must not overlap the `SIDV/NSIDV` fields in `howto_identify`

Example:

```
MPAM_PMG[7:0]=ExtendedID[62:55], MPAM_PARTID[15:0]=ExtendedID[54:39],
MPAM_SP[1:0]=ExtendedID[38:37], MECID[15:0]=UserFlags[31:16]
HWATTR_KIND_0[3:0]=ExtendedID[42:39]
```



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

### **ish\_is\_osh\_DANGER**

#### **Type**

bool

#### **Default value**

true

When set, any transaction that would use the architectural inner shareable domain is converted to use the outer shareable domain.





This parameter should match the equivalent `ish_is_osh` from the PE. If an incompatible value of the `ish_is_osh` parameter is configured for the PE and the SMMU, data coherency may be compromised.



All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

### **legacy\_tz\_en**

#### **Type**

bool

#### **Default value**

false

The default value of the `legacy_tz_en` pin:

**0**

RME is enabled

**1**

RME is disabled

### **list\_of\_ns\_sid\_high\_at\_bitpos0**

#### **Type**

string

#### **Default value**

{}()

A comma-separated list of values to bitwise OR into each Non-secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems. The effective value for any PCIe-RC ports must be the same for non-secure and realm. See `list_of_r_sid_high_at_bitpos0`.

The empty string corresponds to all 0s.

### **list\_of\_pcie\_mode**

#### **Type**

string

**Default value**

""

A comma-separated list of ranges of ports that represent TBUs that are attached to PCIe Root Complexes (PCIe-RC). A single PCIe-RC might use several TBUs and stripe accesses across them. The attribute handling for these TBUs is slightly different in that if the PCIe transaction is NoSnoop and the output attributes of the translation would be weaker than `INC-ONC-OSH` then the output is forced to `INC-ONC-OSH`.

`INC-ONC-OSH` == "inner normal non-cacheable, out normal non-cacheable, outer shared"

**list\_of\_pcie\_rc****Type**

string

**Default value**

""

This is a list of ports that are connected to PCIe Root Complex (PCIe-RC) by a protocol called DTI-ATS. This port is used to transport ATS and PRI Requests to the SMMU from the PCIe-RC.

In the real hardware, the PCIe-RC uses this port for ATS/PRI, and the actual transactions go through separate TBUs. In the model, this port can accept actual transactions as well. However, in the model, then the ATC Invalidates and the PRI Responses need to be transferred over the corresponding `pvbus_id_routed_m` port as DTI-ATS is bidirectional, but PVBus is not.

**list\_of\_r\_sid\_high\_at\_bitpos0****Type**

string

**Default value**

""

A comma-separated list of values to bitwise OR into each Realm StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID. This only has an effect for r1 and later.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems. The effective value for any PCIe-RC ports must be the same for non-secure and realm. See `list_of_ns_sid_high_at_bitpos0`.

The empty string corresponds to all 0s.

"use-ns" can be used to apply the `list_of_ns_sid_high_at_bitpos0` values instead.

**list\_of\_s\_sid\_high\_at\_bitpos0****Type**

string

## Default value

""

A comma-separated list of values to bitwise OR into each Secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

The empty string corresponds to all 0s.

use-ns can be used to apply the `list_of_ns_sid_high_at_bitpos0` values instead.

## mec\_attribute\_transform

### Type

string

### Default value

""



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MEC is supported, this is applied to *all* downstream transactions to transport the MEC information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"UserFlags[31:16]=MECID[15:0]"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/MasterID/UserFlags.

RHS Symbols:

- MECID

Any bits with no transform are unchanged.

Attribute transforms applied before this:



Note

- for client transactions `output_attribute_transform / output_attribute_transform_for_NoStreamID`.
- for table walks `tw_qs_attribute_transform`.
- for MSIs `msi_attribute_transform`.
- if MPAM is enabled `mpam_attribute_transform`.

**mpam\_attribute\_transform****Type**

string

**Default value**

"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID,  
ExtendedID[38]=MPAM\_NS"

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MPAM is supported, this is applied to *all* downstream transactions to transport the MPAM information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID,  
ExtendedID[38]=MPAM_SP[0]"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/MasterID/UserFlags.

RHS Symbols:

- MPAM\_PARTID
- MPAM\_PMG
- MPAM\_NS
- MPAM\_SP
- numeric literals

Any bits with no transform are unchanged.

**Note**

- attribute transforms applied before this:
  - for client transactions `output_attribute_transform / output_attribute_transform_for_NoStreamID`.
  - for table walks `tw_qs_attribute_transform`.
  - for MSIs `msi_attribute_transform`.
- `mec_attribute_transform` is applied after this.
- for translated transactions from client devices then `MPAM_NS = ! SEC_SID`.

**msi\_attribute\_transform****Type**

string

**Default value**

"ExtendedID[31:0]=smmu\_msi\_device\_id, ManagerID64[31:0]=0xFFFFffff"

Transform downstream attributes of MSI transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter output attributes of SMMU-generated MSIs. Example:

```
"UserFlags[15:0]=smmu_msi_device_id[31:16],
ManagerID64[15:0]=smmu_msi_device_id[15:0],
ExtendedID=0"
```

LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Outgoing PVBUS transaction attributes

RHS Symbols:

**smmu\_msi\_device\_id**

The value stored in the parameter with the same name

**interrupt\_kind**

The selected bit corresponds to the interrupts listed below

**0/1**

EVENTQ s/ns

**2**

PRIQ

**3/4**

CMD\_SYNC s/ns

**5/6**

GERROR s/ns

**7/8**

PMCG s/ns

**9/10/11**

RAS FHI/ERI/CRI

**12/13**

gpf\_far/gpt\_cfg\_far

**14/15/16/17**

Realm EVENTQ/PRIQ/CMDQ/GERROR

**HWATTR\_KIND\_0**

PBHA information

**Numeric Literals**

Any number. Ex: 0x1234

ExtendedID/ManagerID64/MasterID/UserFlags start with values {0, 0xFFFFffff, 0xFFFFffff, 0} respectively.

Any bits with no transform are unchanged.

This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices.

**Note**

See also `output_attribute_transform` and `enable_device_id_checks`.

---

**Note**

After 11.25 the `interrupt_kind` field was extended to 5 bits. This is strictly a non-backwards compatible change. However, the original 3 bits were insufficient to express all the interrupt kinds that exist.

---

**normalize\_input\_normal\_non\_iWB\_oWB\_to\_iNC\_oNC\_osh\_DANGER****Type**

bool

**Default value**

true

When set, use Inner Non Cacheable, Outer Non Cacheable, Outer Shareable for any upstream transaction that would use any of the following attributes:

- Normal Non-cacheable Bufferable
- Normal Non-cacheable Non-bufferable
- Write-through

**Note**

This parameter should match the equivalent configuration from the PE. If an incompatible value of this parameter is configured for the PE and the SMMU, data coherency may be compromised.

All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

### **ns\_gbpa\_abort\_init**

#### **Type**

bool

#### **Default value**

false

The default value of the tie off signal `ns_gbpa_abort_init`

### **number\_of\_ports**

#### **Type**

unsigned

#### **Default value**

1

The number of port pairs that the SMMU has.

### **output\_attribute\_transform**

#### **Type**

string

#### **Default value**

"ExtendedID[31:0]=DeviceID"

Transform downstream attributes of StreamID transactions.

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID,
ManagerID64[10]=ManagerID64[11], ManagerID64[11]=ManagerID64[10]"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**  
Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**DeviceID**  
StreamID + translated\_device\_id\_base

**StreamID / SubstreamID / SEC\_SID / SEC\_SID\_bit\_1 / SIDV / SSV**  
Architectural information. See parameter `howto_identify` for more information.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**  
Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

**st1PBHA / st2PBHA**  
Page Based Hardware Attributes from leaf descriptors (zero if unused).

**STE\_IMPDEF1**  
STE[127:116]

**HWATTR\_KIND\_0**  
PBHA information

**Numeric Literals**  
Any number. Ex: 0x1234

The `streamID` has had `ns_sid_high/s_sid_high/r_sid_high` ORred into it for the appropriate TBU.



- `mpam_attribute_transform` and `mec_attribute_transform` are applied in order after this.
- See also `output_attribute_transform_for_NoStreamID` for NoStreamID transactions.

**output\_attribute\_transform\_for\_NoStreamID**

**Type**  
string

**Default value**  
"ExtendedID[31:0]=0, ExtendedID[32]=1"



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

Transform downstream attributes of NoStreamID transactions.



- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=0, UserFlags[31]=1, UserFlags[19:0]=0,
ManagerID64[10]=ManagerID64[11],
ManagerID64[11]=ManagerID64[10] ManagerID64[9:6]=HWATTR_KIND_0"
```

RHS/LHS Symbols: \* ExtendedID/ManagerID64/MasterID/UserFlags: incoming/outgoing attributes.

RHS Symbols: \* SIDV = 0, nSIDV = 1 (fixed values to indicate NoStreamID) \* PAS \*  
HWATTR\_KIND\_0

Any bits with no transform are unchanged.



Note

- mpam\_attribute\_transform and mec\_attribute\_transform are applied in order after this.
- see also output\_attribute\_transform for StreamID transactions.

## output\_id\_routed\_transform

### Type

string

### Default value

"Address[27:12]=StreamID[15:0], PAS=SSD"



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

The SMMU generates the following ID-routed transaction on the pvbus\_id\_routed\_m bus:

- ATC Invalidate
- PRI Response

This parameter controls how the SMMU should express:

- the StreamID
- the Trusted (T) bit

The value is a comma-separated list of assignments:

```
Address[27:12]=StreamID[15:0], ExtendedID[60]=T, ExtendedID[15:0]=StreamID[31:16]
```

Address bits[11:0] cannot be used.

The LHS can be one of:

- PAS
- MasterID / ManagerID64 / ExtendedID / UserFlags
- Address

The RHS can be one of:

- a numeric constant
- SSD
- T or negative version nT
- StreamID

For realm (or 'Trusted') transactions, then  $ssd=0b11$ ,  $T=1$ ,  $nT=0$ . For non-secure (or 'Non-Trusted') transactions, then  $ssd=0b01$ ,  $T=0$ ,  $nT=1$ .

### **prefetch\_only\_requests**

#### **Type**

unsigned

#### **Default value**

0

The simulator supports 'prefetch-only' DMI requests, which can occur at any time and for any reason and are intended to be invisible to the end execution of the model and to the user.

#### **0**

deny all prefetch-only requests

#### **1**

- use debug requests for any page table walks
  - form and use debug TLB/cache entries
  - any faults will not record, but deny the prefetch request

#### **2**

- treat prefetch-only requests like normal transactions
  - use normal page table walk transactions
  - use and form normal TLB/cache entries
  - faults will alter the programmer-visible state of the SMMU

0 is the safest.

1 treats the access like a debug request and requires that debug page table walks are treated correctly downstream. Any descriptors that need HTTU to allow the transaction to proceed will fail the request.

2 is dangerous, it uses real transactions and reports faults that are unphysical. Real transactions can be `wait()`ed and this disobeys the SystemC spec for `get_direct_mem_ptr()`.

### **rme\_10gpt\_entry\_covers\_log2size\_in\_bytes**

#### **Type**

uint32\_t

#### **Default value**

30

Each LOGPT entry covers:

```
2**rme_10gpt_entry_covers_log2size_in_bytes
```

bytes of address space.

The valid values for this parameter are: 30, 34, 36, 39

This parameter is reported in an encoded format as the read-only field:

```
SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ
```

This parameter can be overridden by the port `10gptsz_s` when sampled on negedge of reset.

### **s\_gbpa\_abort\_init**

#### **Type**

bool

#### **Default value**

false

The default value of the tie off signal `s_gbpa_abort_init`

### **sec\_override**

#### **Type**

bool

#### **Default value**

false

The IMP DEF port `sec_override` controls whether some of the registers are accessible to secure or non-secure/realms transactions. This parameter is the default value assumed for that port if the port is not driven by a signal.

### **seed**

#### **Type**

uint32\_t

**Default value**

0x12345678

Used to seed the pseudo-random number generator that the SMMU model uses.

**size\_of\_cd\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding CD structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_dpttlb****Type**

uint32\_t

**Default value**

0

The number of entries in the DPT TLB. If this is zero then it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_l1cd\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding L1CD descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_l1ste\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding L1STE descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_ste\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding STE structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_tlb****Type**

uint32\_t

**Default value**

0

The number of entries in the TLB. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**smmu\_msi\_device\_id****Type**

uint32\_t

**Default value**

0

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID.

See parameters `msi_attribute_transform` and `enable_device_id_checks`.

**sup\_btm****Type**

bool

**Default value**

true

The default value of the register field `SMMU_IDR0.BTM` and `SMMU_ROOT_IDR0.BGPTM` (when supported). This enables Broadcast TLB maintenance.

**sup\_cohacc****Type**

bool

**Default value**

true

The default value of the register `SMMU_IDR0.COHAAC`

### **sup\_httu**

#### **Type**

bool

#### **Default value**

true

The initial value of the `sup_httu` port. See the port description for `sup_httu`.

### **sup\_oas**

#### **Type**

unsigned

#### **Default value**

6

The hardware has an input port `sup_oas[2:0]` that indicates what output address size (OAS) the system has. This is sampled at reset.

The model does not have this port as it is expected to be a constant for the system and not to change. Instead it is just a parameter.

The allowed values are:

**0**

32 bits

**1**

36 bits

**2**

40 bits

**3**

42 bits

**4**

44 bits

**5**

48 bits

**6**

52 bits

### **sup\_sev**

#### **Type**

bool

**Default value**

true

The default value of the register `SMMU_IDR0.SEV`

**tlb\_when\_do\_f\_tlb\_conflict\_on\_overlap****Type**

unsigned

**Default value**

0

If a TLB entry is created by a walk and it overlaps an existing entry, there are some architectural situations where the result is known. For all others, then an implementation is allowed to use an **UNPREDICTABLE** combination of the two entries, or it can generate `F_TLB_CONFLICT`:

**0**

never generate

**1**

sometimes generate

**2**

always generate

Conflicts between global and non-global entries are not detected by the model.

**translated\_device\_id\_base****Type**

uint32\_t

**Default value**

0

When appropriately enabled, assume that client device accesses are translated to a DeviceID as seen by the GIC of:

```
StreamID + translated_device_id_base
```

See parameter `output_attribute_transform` and `enable_device_id_checks`.

**tw\_qs\_attribute\_transform****Type**

string

**Default value**{}  
(0)

Transform downstream attributes of table walk and queue transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**HWATTR\_KIND\_0**

PBHA information

**kind**

Transaction kind

- For a read:

**0/1**

L1STE/STE

**2/3**

L1CD/CD

**4/5**

S1/S2 TTD (including CAS)

**6**

CMDQ

**7**

VMS

**11/12**

LOGPT/L1GPT

**13/14**

LODPT/L1DPT

- For a write:

**0**

EVENTQ

**1**

PRIQ



ExtendedID/ManagerID64/MasterID/UserFlags start with values {0, 0xFFFFffff, 0xFFFFffff, 0} respectively.

Any bits with no transform are unchanged.



See also `output_attribute_transform` and `msi_attribute_transform`.

---

## **version**

### **Type**

string

### **Default value**

"r0p0"

The version of this product.

Valid values are:

- r0p0
- r1p0

## **wait\_cmdq\_ticks**

### **Type**

uint64\_t

### **Default value**

0

This is the time to wait before doing something on the command queue. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffff)) - 1]$ .

## **wait\_eventq\_ticks**

### **Type**

uint64\_t

### **Default value**

0

This is the time to wait before doing something on the event queue. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffff)) - 1]$ .

**wait\_misc\_async\_actions\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing an async action that could be delayed is performed. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

**wait\_msi\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before sending an MSI. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

**wait\_pri\_req\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before processing a PRI Request. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

**wait\_pri\_resp\_ticks****Type**

uint64\_t

**Default value**

1

This is the time to wait before sending a PRI Response back to the PCIe subsystem. When a PRI Response is an auto-response then the ATC might immediately make a new ATS request, that immediately fails, immediately makes a PRI Request, or auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

### 3.10.65 MMU\_S3

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1107: IP revisions support**

Revision	Quality level
r0p0	Full support
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Changes in 11.29.19

Parameters added:

- `hide_warning_ACCESSEN_GPCEN_set_to_1_in_a_single_write`

#### Limitations

- No power control
- AMBA stash operations, destructive read and destructive hint operations are not supported on PVBUS and also are not supported by the device.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMCB_CBEID0` fields. The PMU is intended for demonstration purposes only and for driver development.
- Limited RAS support
- The SYSCO interface is not implemented
- The low power interface is not implemented
- The IMP DEF MPAM register file is not implemented. This controls how the internal resources of the MMU-S3 are partitioned.
- `tcu_sid[31:0]` is not modelled, instead the parameter `smmu_msi_device_id` is used.
- The HWATTR side-band signal is available by configuring `output_attribute_transform`, `msi_attribute_transform` and `tw_qs_attribute_transform` with `HWATTR_KIND_0`. For some transactions HWATTR comes from the `SMMU_S_AGBPA[3:0]`. In the hardware, this register has an 'Update' bit[31] that should be written as 1 and will be turned to zero when all transactions using the old value have completed. The model does not implement this behavior and the Update bit is **RAZ/WI**.
- Any configuration parameter listed in the TRM but not shown in this file is not supported.
- `TCU_CTRL_AUX0-55` registers are modeled but unlike the TRM, reset values for these registers are 0. There is no functionality associated with these registers.
- There is no functionality associated with `TCU_ROOT_CTRL` register field `DIS_DVM`. Note: If `DIS_DVM` is set to 1, an error message is thrown which can be disabled by setting `all_error_messages_through_trace` to true.

- The model does not support the `eventoack` signal. In integration mode, the value of the field `eventoack` (bit 0) in `ITIN_PIU` will match the value of the `evento` signal.
- There is no functionality associated with `TCU_NODE_STATUS` register field `ATSv`, which indicates whether the node implements DTI-ATSv4.

## Notes

- A bad configuration renders the model inactive.
- Some configurations can be adjusted by configuration pins. These are only sampled at the negative edge of the reset pin. If you want to use these pins then you must drive them before sending a negative edge on the reset pin. During simulation reset the component driving them must also drive this transition again.
- Debug reads to the registers do not disturb the state.
- Writes to registers with Update flags, including debug writes, are ignored if the Update flag is already set to one.
- Debug and real accesses to the registers must be 32 bits or 64 bits.
- The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping used is remembered by the bus infrastructure and is used for subsequent sufficiently similar transactions without requiring intervention from the SMMU model, and is not traced, for instance.

The range of addresses that the mapping is valid for is determined by the SMMU model, depending on architectural and model-implementation details. However, as it is unaware of any sign-extension or the mapping that the SoC does, the SoC is responsible for subsequently narrowing the range of addresses for which this mapping is valid. Typically, this is done automatically when using `PVBusMapper`.

- The hardware is a distributed SMMU and is divided into:
  - A single Translation Control Unit (TCU)
    - Has a port for the programming interface of the SMMU
    - Receives DVM messages
    - Does all the page walking, queue manipulation, etc.
  - One or more Translation Bus Units (TBUs)
    - Translate transactions from upstream (client) device into downstream transactions.
  - Zero or more connections to PCIe Root Complexes (PCIe-RCs)
  - There can be a total of 62 TBUs and PCIe-RCs attached.
  - An interconnect connecting the TBUs, PCIe-RCs to the TCU.
- A PCIe-RC has one connection to the TCU to make ATS requests but the PCIe-RC uses one or more TBUs to transform the transactions and pass them to the memory system. In the model, those TBUs are configured by the parameter `list_of_pcie_mode`. The SMMU does not know which TBUs a particular PCIe-RC is attached to.
- The TBU ORs a value into each StreamID that it receives. In the model, this is configured by the parameters:

- `list_of_s_sid_high_at_bitpos0`
- `list_of_ns_sid_high_at_bitpos0`
- The TCU, TBU, and the interconnect are all represented by this single model component.
- In the model, a pair of ports `tbs_pvbus_s[i]/tbm_pvbus_m[i]` represent a TBU 'i' or the `tbs_pvbus_s[i]` represents an incoming connection for a PCIe-RC. The corresponding reverse connection from the TCU to the PCIe-RC is by a special bus called `pvbus_id_routed_m` that is used to transport ATC Invalidates to the PCIe-RC.
  - In order to reduce system construction complexity the `tbs_pvbus_s[i]/tbm_pvbus_m[i]` also acts as a TBU so that the PCIe-RC need not separate its normal transactions and its ATS requests.
  - However, ATC Invalidates are only sent to a port which appears in `list_of_pcie_rc`. It should be uniquely decoded to a single port based on `list_of_ns_sid_high_at_bitpos0` and those ATC Invalidates must be routed to the correct PCIe subsystem in order to invalidate the cache of ATS Response in the subsystem. Thus all TBUs that a PCIe-RC uses must have a unique reverse mapping from stream id to port.
- The `pin_sup_oas` is not supported, instead it is a parameter as it is assumed that it would be tied to a fixed value in any specific platform.
- The hardware only has a single cacheability attribute for input transactions, but PVBUS transports both inner and outer cacheability.
- For non-PCIe-mode TBUs (i.e. their index does not appear in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - For non-cache maintenance operations:
    - If the input attribute is any type of device then it is well defined as being outer-shared and Device-nGnRnE or Device-nGnRE. There is no support for Gathering or Reordering.
    - If the outer cacheable input attribute is normal then if it is Write-back then this is converted to Inner Write-back Outer Write-back (iWB-oWB) with the desired shareability. No Transient hint is supported and is always treated as non-transient.
    - This leaves all other normal memory types that are mapped to Inner Normal Non-cacheable, Outer Normal Non-cacheable outer shared (iNC-oNC-osh).
- Thus the upstream devices must present the cacheability in the *outer* cacheability attribute on PVBUS if it is cacheable. If it is a device type then both the inner and outer attributes must be set to the same. If it is iNC-oNC-osh then it must be presented as such.
- For PCIe-mode TBUs (i.e. their index appears in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - Input transactions are from PCIe and the only indication of the memory type is in the NoSnoop bit of the transaction. No shareability is transported.
    - NoSnoop interpreted as iNC-oNC-osh
    - ! NoSnoop interpreted as iWB-oWB-ish (note inner shareable)
  - If a NoSnoop transaction has an attribute transform applied to it and the result of the transform is weaker than iNC-oNC-osh then it is forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it is forced to iNC-oNC-osh. However, if the page tables transformed it to a device type then, as all device types are stronger than iNC-oNC-osh then it exits the SMMU as the device type.

- In the model, transactions are classified by their incoming memory attributes as to whether they are NoSnoop or not and then normalized appropriately:
  - iWB-oWB-any-shareability transactions are interpreted as ! NoSnoop and therefore are normalized to iWB-oWB-ish.
  - Anything else is considered NoSnoop and therefore is normalized to iNC-oNC-osh.
- Translated accesses also suffer the same interpretation to determine NoSnoop and how they are normalized. Thus they could enter the system with attributes different to if they were Untranslated Accesses translated by normal means.
- It is expected that there is a component downstream of the SMMU that is aware of the system address map and will override the memory type to 'device' for any transaction that accesses a peripheral.
- The hardware has a single cacheability on input and, for transactions that are neither cache-maintenance operations nor PCIe transactions, normalizes the input to an architectural form before performing the SMMUv3 architectural transform:
  - Any device type is left untouched (the input can only represent Device-nGnRE and Device-nGnRnE).
  - If the input is Write-back (WB) then it is normalized to iWB-oWB with the incoming shareability.
  - If the input is anything else it is normalized to iNC-oNC-osh.
- The model accepts full architectural attributes of two levels of cacheability and so has to decide how to interpret this in terms of the hardware. For transactions that are not cache maintenance operations, the model replicates the outer attribute into the inner attribute and then performs the normalization that the hardware does.
- The hardware normalizes the architectural output attributes and outputs a single level of cacheability and a user flag (OC) specifying if the architectural attributes were cacheable in the outer cacheable domain. If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. If the original transaction was NoSnoop, then any weaker memory type is strengthened to iNC-oNC-osh so apply the following transform:

```

if      iWB-oWB-nsh/ish/osh      then output WB-nsh/ish/osh, OC = 1
else if i (NC/WT/WB) -o (WB/WT) then output NC-Sys,      OC = 1
else if i (NC/WT/WB) -oNC        then output NC-Sys,      OC = 0
else if Device-(GRE/nGRE/nGnRE) then output DV-Sys,      OC = 0
else                               output SO-Sys,      OC = 0

```

- The model only normalizes according to PCIe but otherwise leaves the architectural attributes intact on the output bus.
- MSIs are issued on the `qtw_pvbuss_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `MasterID=0xFFFFffff`.

In the hardware, there is no way of distinguishing Event Queue writes from MSI writes, however, this provides a mechanism that if the model system needs to distinguish then it can. MPAM and MEC attributes are provided by the parameters:

- `mpam_attribute_transform`

- `mec_attribute_transform` (not all versions support MEC)
- Note that some older versions of MMU-S3 documentation/RTL referred to the `ns_gbpa_abort_init` and `s_gbpa_abort_init` tie-offs as `sup_ns_gbpa_abort_rst` and `sup_s_gbpa_abort_rst`.
- The model supports architectural features and registers matching `r0p0-00eac0` and `r1p0-00eac0`.

## Security State Determination (SSD)

The model uses the term SSD to mean the security state that the transaction, register, structure, or event belongs to. In the SMMUv3 architecture, the sideband signal `SEC_SID` holds this information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one-bit signal. With RME, in the hardware, it was expanded to two bits. To retain backwards compatibility in the model, `SEC_SID` remains as one-bit in the parameter `howto_identify`, but the second bit can be expressed with `SEC_SID_bit_1`, and its negative logic version `nSEC_SID_bit_1`.

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

## Parameters for parsing transaction attributes

Some of the parameters are strings that share a common parsing format for extracting from and placing information into the transaction's attributes.

They can extract and place information into the following fields of a transaction's attributes:

- `ManagerID64` (64 bits)
- `MasterID` (32 bits) (This is the lower 32 bits of `ManagerID64`)
- `ExtendedID` (64 bits)
- `UserFlags` (32 bits)

The string parameter is parsed as a comma-separated list of:

```
lhs_expression=rhs_expression
```

The `lhs_expression` and `rhs_expression` can be an entire symbol, for example:

```
StreamID
```

or a bit, or bit slice:

```
StreamID[16]
```

```
StreamID[31:20]
```

In `rhs_expression`, numeric constants (or slices of numeric constants) are also allowed:

```
StreamID[16]=1
StreamID[16]=10[2]
```

A single symbol might be assigned from multiple non-contiguous arrays of bits from a mix of different RHS symbols:

```
StreamID[16]=1, StreamID[31:30]=ExtendedID[1:0],
StreamID[15:0]=UserFlags[31:16], ...
```

In those cases where a left hand symbol can also appear on the right hand side, it is possible to swap bits and transform the symbol. For example, the following expression swaps bits 0 and 1 of the `ExtendedID`:

```
ExtendedID[0]=ExtendedID[1], ExtendedID[1]=ExtendedID[0]
```

Any bits of the attributes that have no transform specified are retained from the input.

The `lhs_expression` and `rhs_expression` must have the same bit width.

## Iris and MTI instances for MMU\_S3

This model has the following Iris instances:

**Table 3-1109: MMU\_S3 Iris instances**

InstanceName	ComponentName
MMU_S3	MMU_S3
MMU_S3.register_file[0]	PVBusSlave
MMU_S3.service_request_tbu[0]	PVBusSlave
MMU_S3.service_request_tbu[10]	PVBusSlave
MMU_S3.service_request_tbu[11]	PVBusSlave
MMU_S3.service_request_tbu[12]	PVBusSlave
MMU_S3.service_request_tbu[13]	PVBusSlave
MMU_S3.service_request_tbu[14]	PVBusSlave
MMU_S3.service_request_tbu[15]	PVBusSlave
MMU_S3.service_request_tbu[16]	PVBusSlave
MMU_S3.service_request_tbu[17]	PVBusSlave
MMU_S3.service_request_tbu[18]	PVBusSlave
MMU_S3.service_request_tbu[19]	PVBusSlave
MMU_S3.service_request_tbu[1]	PVBusSlave
MMU_S3.service_request_tbu[20]	PVBusSlave
MMU_S3.service_request_tbu[21]	PVBusSlave



InstanceName	ComponentName
MMU_S3.service_request_tbu[22]	PVBusSlave
MMU_S3.service_request_tbu[23]	PVBusSlave
MMU_S3.service_request_tbu[24]	PVBusSlave
MMU_S3.service_request_tbu[25]	PVBusSlave
MMU_S3.service_request_tbu[26]	PVBusSlave
MMU_S3.service_request_tbu[27]	PVBusSlave
MMU_S3.service_request_tbu[28]	PVBusSlave
MMU_S3.service_request_tbu[29]	PVBusSlave
MMU_S3.service_request_tbu[2]	PVBusSlave
MMU_S3.service_request_tbu[30]	PVBusSlave
MMU_S3.service_request_tbu[31]	PVBusSlave
MMU_S3.service_request_tbu[32]	PVBusSlave
MMU_S3.service_request_tbu[33]	PVBusSlave
MMU_S3.service_request_tbu[34]	PVBusSlave
MMU_S3.service_request_tbu[35]	PVBusSlave
MMU_S3.service_request_tbu[36]	PVBusSlave
MMU_S3.service_request_tbu[37]	PVBusSlave
MMU_S3.service_request_tbu[38]	PVBusSlave
MMU_S3.service_request_tbu[39]	PVBusSlave
MMU_S3.service_request_tbu[3]	PVBusSlave
MMU_S3.service_request_tbu[40]	PVBusSlave
MMU_S3.service_request_tbu[41]	PVBusSlave
MMU_S3.service_request_tbu[42]	PVBusSlave
MMU_S3.service_request_tbu[43]	PVBusSlave
MMU_S3.service_request_tbu[44]	PVBusSlave
MMU_S3.service_request_tbu[45]	PVBusSlave
MMU_S3.service_request_tbu[46]	PVBusSlave
MMU_S3.service_request_tbu[47]	PVBusSlave
MMU_S3.service_request_tbu[48]	PVBusSlave
MMU_S3.service_request_tbu[49]	PVBusSlave
MMU_S3.service_request_tbu[4]	PVBusSlave
MMU_S3.service_request_tbu[50]	PVBusSlave
MMU_S3.service_request_tbu[51]	PVBusSlave
MMU_S3.service_request_tbu[52]	PVBusSlave
MMU_S3.service_request_tbu[53]	PVBusSlave
MMU_S3.service_request_tbu[54]	PVBusSlave
MMU_S3.service_request_tbu[55]	PVBusSlave
MMU_S3.service_request_tbu[56]	PVBusSlave
MMU_S3.service_request_tbu[57]	PVBusSlave

InstanceName	ComponentName
MMU_S3.service_request_tbu[58]	PVBusSlave
MMU_S3.service_request_tbu[59]	PVBusSlave
MMU_S3.service_request_tbu[5]	PVBusSlave
MMU_S3.service_request_tbu[60]	PVBusSlave
MMU_S3.service_request_tbu[61]	PVBusSlave
MMU_S3.service_request_tbu[62]	PVBusSlave
MMU_S3.service_request_tbu[63]	PVBusSlave
MMU_S3.service_request_tbu[6]	PVBusSlave
MMU_S3.service_request_tbu[7]	PVBusSlave
MMU_S3.service_request_tbu[8]	PVBusSlave
MMU_S3.service_request_tbu[9]	PVBusSlave
MMU_S3.tbu[0]	PVBusMapper
MMU_S3.tbu[10]	PVBusMapper
MMU_S3.tbu[11]	PVBusMapper
MMU_S3.tbu[12]	PVBusMapper
MMU_S3.tbu[13]	PVBusMapper
MMU_S3.tbu[14]	PVBusMapper
MMU_S3.tbu[15]	PVBusMapper
MMU_S3.tbu[16]	PVBusMapper
MMU_S3.tbu[17]	PVBusMapper
MMU_S3.tbu[18]	PVBusMapper
MMU_S3.tbu[19]	PVBusMapper
MMU_S3.tbu[1]	PVBusMapper
MMU_S3.tbu[20]	PVBusMapper
MMU_S3.tbu[21]	PVBusMapper
MMU_S3.tbu[22]	PVBusMapper
MMU_S3.tbu[23]	PVBusMapper
MMU_S3.tbu[24]	PVBusMapper
MMU_S3.tbu[25]	PVBusMapper
MMU_S3.tbu[26]	PVBusMapper
MMU_S3.tbu[27]	PVBusMapper
MMU_S3.tbu[28]	PVBusMapper
MMU_S3.tbu[29]	PVBusMapper
MMU_S3.tbu[2]	PVBusMapper
MMU_S3.tbu[30]	PVBusMapper
MMU_S3.tbu[31]	PVBusMapper
MMU_S3.tbu[32]	PVBusMapper
MMU_S3.tbu[33]	PVBusMapper
MMU_S3.tbu[34]	PVBusMapper

InstanceName	ComponentName
MMU_S3.tb[35]	PVBusMapper
MMU_S3.tb[36]	PVBusMapper
MMU_S3.tb[37]	PVBusMapper
MMU_S3.tb[38]	PVBusMapper
MMU_S3.tb[39]	PVBusMapper
MMU_S3.tb[3]	PVBusMapper
MMU_S3.tb[40]	PVBusMapper
MMU_S3.tb[41]	PVBusMapper
MMU_S3.tb[42]	PVBusMapper
MMU_S3.tb[43]	PVBusMapper
MMU_S3.tb[44]	PVBusMapper
MMU_S3.tb[45]	PVBusMapper
MMU_S3.tb[46]	PVBusMapper
MMU_S3.tb[47]	PVBusMapper
MMU_S3.tb[48]	PVBusMapper
MMU_S3.tb[49]	PVBusMapper
MMU_S3.tb[4]	PVBusMapper
MMU_S3.tb[50]	PVBusMapper
MMU_S3.tb[51]	PVBusMapper
MMU_S3.tb[52]	PVBusMapper
MMU_S3.tb[53]	PVBusMapper
MMU_S3.tb[54]	PVBusMapper
MMU_S3.tb[55]	PVBusMapper
MMU_S3.tb[56]	PVBusMapper
MMU_S3.tb[57]	PVBusMapper
MMU_S3.tb[58]	PVBusMapper
MMU_S3.tb[59]	PVBusMapper
MMU_S3.tb[5]	PVBusMapper
MMU_S3.tb[60]	PVBusMapper
MMU_S3.tb[61]	PVBusMapper
MMU_S3.tb[62]	PVBusMapper
MMU_S3.tb[63]	PVBusMapper
MMU_S3.tb[6]	PVBusMapper
MMU_S3.tb[7]	PVBusMapper
MMU_S3.tb[8]	PVBusMapper
MMU_S3.tb[9]	PVBusMapper

This model has the following MTI trace components:

**Table 3-1110: MMU\_S3 MTI instances**

InstanceName	ComponentName
MMU_S3	MMU_S3
MMU_S3.register_file[0]	PVBusSlave
MMU_S3.service_request_tbu[0]	PVBusSlave
MMU_S3.service_request_tbu[10]	PVBusSlave
MMU_S3.service_request_tbu[11]	PVBusSlave
MMU_S3.service_request_tbu[12]	PVBusSlave
MMU_S3.service_request_tbu[13]	PVBusSlave
MMU_S3.service_request_tbu[14]	PVBusSlave
MMU_S3.service_request_tbu[15]	PVBusSlave
MMU_S3.service_request_tbu[16]	PVBusSlave
MMU_S3.service_request_tbu[17]	PVBusSlave
MMU_S3.service_request_tbu[18]	PVBusSlave
MMU_S3.service_request_tbu[19]	PVBusSlave
MMU_S3.service_request_tbu[1]	PVBusSlave
MMU_S3.service_request_tbu[20]	PVBusSlave
MMU_S3.service_request_tbu[21]	PVBusSlave
MMU_S3.service_request_tbu[22]	PVBusSlave
MMU_S3.service_request_tbu[23]	PVBusSlave
MMU_S3.service_request_tbu[24]	PVBusSlave
MMU_S3.service_request_tbu[25]	PVBusSlave
MMU_S3.service_request_tbu[26]	PVBusSlave
MMU_S3.service_request_tbu[27]	PVBusSlave
MMU_S3.service_request_tbu[28]	PVBusSlave
MMU_S3.service_request_tbu[29]	PVBusSlave
MMU_S3.service_request_tbu[2]	PVBusSlave
MMU_S3.service_request_tbu[30]	PVBusSlave
MMU_S3.service_request_tbu[31]	PVBusSlave
MMU_S3.service_request_tbu[32]	PVBusSlave
MMU_S3.service_request_tbu[33]	PVBusSlave
MMU_S3.service_request_tbu[34]	PVBusSlave
MMU_S3.service_request_tbu[35]	PVBusSlave
MMU_S3.service_request_tbu[36]	PVBusSlave
MMU_S3.service_request_tbu[37]	PVBusSlave
MMU_S3.service_request_tbu[38]	PVBusSlave
MMU_S3.service_request_tbu[39]	PVBusSlave
MMU_S3.service_request_tbu[3]	PVBusSlave
MMU_S3.service_request_tbu[40]	PVBusSlave
MMU_S3.service_request_tbu[41]	PVBusSlave

InstanceName	ComponentName
MMU_S3.service_request_tbu[42]	PVBusSlave
MMU_S3.service_request_tbu[43]	PVBusSlave
MMU_S3.service_request_tbu[44]	PVBusSlave
MMU_S3.service_request_tbu[45]	PVBusSlave
MMU_S3.service_request_tbu[46]	PVBusSlave
MMU_S3.service_request_tbu[47]	PVBusSlave
MMU_S3.service_request_tbu[48]	PVBusSlave
MMU_S3.service_request_tbu[49]	PVBusSlave
MMU_S3.service_request_tbu[4]	PVBusSlave
MMU_S3.service_request_tbu[50]	PVBusSlave
MMU_S3.service_request_tbu[51]	PVBusSlave
MMU_S3.service_request_tbu[52]	PVBusSlave
MMU_S3.service_request_tbu[53]	PVBusSlave
MMU_S3.service_request_tbu[54]	PVBusSlave
MMU_S3.service_request_tbu[55]	PVBusSlave
MMU_S3.service_request_tbu[56]	PVBusSlave
MMU_S3.service_request_tbu[57]	PVBusSlave
MMU_S3.service_request_tbu[58]	PVBusSlave
MMU_S3.service_request_tbu[59]	PVBusSlave
MMU_S3.service_request_tbu[5]	PVBusSlave
MMU_S3.service_request_tbu[60]	PVBusSlave
MMU_S3.service_request_tbu[61]	PVBusSlave
MMU_S3.service_request_tbu[62]	PVBusSlave
MMU_S3.service_request_tbu[63]	PVBusSlave
MMU_S3.service_request_tbu[6]	PVBusSlave
MMU_S3.service_request_tbu[7]	PVBusSlave
MMU_S3.service_request_tbu[8]	PVBusSlave
MMU_S3.service_request_tbu[9]	PVBusSlave
MMU_S3.tbu[0]	PVBusMapper
MMU_S3.tbu[10]	PVBusMapper
MMU_S3.tbu[11]	PVBusMapper
MMU_S3.tbu[12]	PVBusMapper
MMU_S3.tbu[13]	PVBusMapper
MMU_S3.tbu[14]	PVBusMapper
MMU_S3.tbu[15]	PVBusMapper
MMU_S3.tbu[16]	PVBusMapper
MMU_S3.tbu[17]	PVBusMapper
MMU_S3.tbu[18]	PVBusMapper
MMU_S3.tbu[19]	PVBusMapper

InstanceName	ComponentName
MMU_S3.tbu[1]	PVBusMapper
MMU_S3.tbu[20]	PVBusMapper
MMU_S3.tbu[21]	PVBusMapper
MMU_S3.tbu[22]	PVBusMapper
MMU_S3.tbu[23]	PVBusMapper
MMU_S3.tbu[24]	PVBusMapper
MMU_S3.tbu[25]	PVBusMapper
MMU_S3.tbu[26]	PVBusMapper
MMU_S3.tbu[27]	PVBusMapper
MMU_S3.tbu[28]	PVBusMapper
MMU_S3.tbu[29]	PVBusMapper
MMU_S3.tbu[2]	PVBusMapper
MMU_S3.tbu[30]	PVBusMapper
MMU_S3.tbu[31]	PVBusMapper
MMU_S3.tbu[32]	PVBusMapper
MMU_S3.tbu[33]	PVBusMapper
MMU_S3.tbu[34]	PVBusMapper
MMU_S3.tbu[35]	PVBusMapper
MMU_S3.tbu[36]	PVBusMapper
MMU_S3.tbu[37]	PVBusMapper
MMU_S3.tbu[38]	PVBusMapper
MMU_S3.tbu[39]	PVBusMapper
MMU_S3.tbu[3]	PVBusMapper
MMU_S3.tbu[40]	PVBusMapper
MMU_S3.tbu[41]	PVBusMapper
MMU_S3.tbu[42]	PVBusMapper
MMU_S3.tbu[43]	PVBusMapper
MMU_S3.tbu[44]	PVBusMapper
MMU_S3.tbu[45]	PVBusMapper
MMU_S3.tbu[46]	PVBusMapper
MMU_S3.tbu[47]	PVBusMapper
MMU_S3.tbu[48]	PVBusMapper
MMU_S3.tbu[49]	PVBusMapper
MMU_S3.tbu[4]	PVBusMapper
MMU_S3.tbu[50]	PVBusMapper
MMU_S3.tbu[51]	PVBusMapper
MMU_S3.tbu[52]	PVBusMapper
MMU_S3.tbu[53]	PVBusMapper
MMU_S3.tbu[54]	PVBusMapper

InstanceName	ComponentName
MMU_S3.tbv[55]	PVBusMapper
MMU_S3.tbv[56]	PVBusMapper
MMU_S3.tbv[57]	PVBusMapper
MMU_S3.tbv[58]	PVBusMapper
MMU_S3.tbv[59]	PVBusMapper
MMU_S3.tbv[5]	PVBusMapper
MMU_S3.tbv[60]	PVBusMapper
MMU_S3.tbv[61]	PVBusMapper
MMU_S3.tbv[62]	PVBusMapper
MMU_S3.tbv[63]	PVBusMapper
MMU_S3.tbv[6]	PVBusMapper
MMU_S3.tbv[7]	PVBusMapper
MMU_S3.tbv[8]	PVBusMapper
MMU_S3.tbv[9]	PVBusMapper

## Ports for MMU\_S3

**Table 3-1111: Ports**

Name	Protocol	Type	Description
axi_stream_msi_addr_to_match_s	Value_64	Slave	Any SMMU-originated MSI which exactly matches the address in this port will be sent through the AXI stream port axi_stream_msi_m which is usually connected to the GIC through axi_stream_msi_s. As MSIs are 32 bit aligned then if bits[1:0] != 0 or the address is above OAS then this is effectively disabled. NOTE that the model does not support tcu_sid[31:0] which is the MSI DeviceID to send on axi_stream_msi_m. Instead the parameter smmu_msi_device_id is used. See also the parameters: axi_stream_msi_TID and axi_stream_msi_TDEST. The default value of this port is set by the parameter axi_stream_msi_addr_to_match. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
axi_stream_msi_m	PVBus	Master	Manager port used for sending SMMU originated MSIs directly to the GIC
clk_in	ClockSignal	Slave	Clock signal (in RTL aclk) This is a clock time-base used by the TCU to spread some of its processing over time, if enabled by the wait_* parameters. The clock must always be connected.
cmd_sync_irpt_ns	Signal	Master	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of SIG_IRQ.
cmd_sync_irpt_r	Signal	Master	Pulsed interrupt output signal for realm CMD_SYNC having a completion signal of SIG_IRQ. This pin exists in r0 but is tied off to 0. It is implemented in r1.
cmd_sync_irpt_s	Signal	Master	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
event_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.

Name	Protocol	Type	Description
event_q_irpt_r	Signal	Master	Pulsed interrupt output signal for the realm event queue becoming non-empty. This pin exists in r0 but is tied off to 0. It is implemented in r1.
event_q_irpt_s	Signal	Master	Pulsed interrupt output signal for the secure event queue becoming non-empty.
evento	Signal	Master	Event signal This aligns with the eventoreq signal on the RTL. The eventock signal is not supported.
global_irpt_ns	Signal	Master	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signalling an error.
global_irpt_r	Signal	Master	Pulsed interrupt output signal for realm SMMU_R_GERROR(N) signalling an error. This pin exists in r0 but is tied off to 0. It is implemented in r1.
global_irpt_s	Signal	Master	Pulsed interrupt output signal for secure SMMU_S_GERROR(N) signalling an error.
gpf_far	Signal	Master	An error becomes active in SMMU_ROOT_GPF_FAR.
gpt_cfg_far	Signal	Master	An error becomes active in SMMU_ROOT_GPT_CFG_FAR.
identify	SMMUv3AEMIdentifyProtocol	Master	Map the transaction to the tuple (StreamID, SubStreamID, SubStreamIDValid, SSD) The StreamID that is produced by the implementation of this protocol is not the final StreamID. The final StreamID is produced by using the list_of_ns_sid_high_at_bitpos0/list_of_s_sid_high_at_bitpos0 parameter to map the StreamID based on the upstream port index. Also see the parameter howto_identify which can replace the functionality of this port under certain circumstances.
logptsz_s	Value	Slave	RME: This is a four bit signal that encodes the region size that a single LOGPT entry covers. The default value of this port is derived from the parameter rme_logpt_entry_covers_log2size_in_bytes which is in a different format to the port. If a valid value is driven then it will be put in the field SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ. The port uses the same encoding as the field. If an invalid value is driven to this port and legacy_tz_en is low then the value is reported in the SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ and then all transactions will fault with a GPT Configuration fault (gpt_cfg_far). This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
legacy_tz_en	Signal	Slave	Tie this high to get non-RME behaviour. On the real hardware, then each of the TCUs and the TBUs have a legacy_tz_en and they must all be driven to the same value. In the model, we only have a single version of this pin. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
ns_gbpa_abort_init	Signal	Slave	This port is an Non-secure global bypass. The ns_gbpa_abort_init signal sets the reset value of SMMU_GBPA.ABORT: 0 - On reset, do not abort all incoming transactions. 1 - On reset, abort all incoming transactions. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
pri_q_irpt_ns	Signal	Master	Pulsed interrupt output signal for the non-secure PRI queue becoming non-empty.



Name	Protocol	Type	Description
pri_q_irpt_r	Signal	Master	Pulsed interrupt output signal for the realm PRI queue becoming non-empty. This pin exists in r0 but is tied off to 0. It is implemented in r1.
prog_pvbus_s	PVBus	Slave	Register subordinate port (in RTL PROG)
pvbus_id_routed_m[62]	PVBus	Master	This is a special "id-routed" port for transmitting ATC invalidates upstream into the PCIe EndPoints, it is not a normal bus. See parameter output_id_routed_transform. The FastModels ATC Invalidate and PRI Response protocol specifies how to route and deal with this port.
qtw_pvbus_m	PVBus	Master	Manager port used for Table Walks, MSIs and Queue access. Note that although this is a manager port, it can still receive DVM messages.
s_gbpa_abort_init	Signal	Slave	This port is an secure global bypass. The s_gbpa_abort_init signal sets the reset value of SMMU_S_GBPA.ABORT: 0 - On reset, do not abort all incoming transactions. 1 - On reset, abort all incoming transactions. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sec_override	Signal	Slave	Allow certain registers to be accessible to non-secure accesses from reset, as described in the TCU_SCR register. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_btm	Signal	Slave	System supports BTM and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If BTM (Broadcast Table Maintenance) is not supported then DVM messages will be ignored. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_cohacc	Signal	Slave	System supports COHACC and will be reflected in the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_httu	Signal	Slave	System supports HTTU and will be reflected in the value of SMMU_IDR0.HTTU: - 0b00 (HTTU not supported) if sup_httu is 0 - 0b10 (update of AF and Dirty flags supported) if sup_httu is 1 The default value for this pin is 1. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_sev	Signal	Slave	System supports SEV and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
tbm_pvbus_m[62]	PVBus	Master	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered tbs_pvbus_s[] port.
tbs_pvbus_s[62]	PVBus	Slave	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
tbu_crit_err[62]	Signal	Master	Critical error. This cannot occur in the model except by using an Integration Register to generate it.
tbu_pmu_irpt[62]	Signal	Master	TBU Performance Monitoring Unit interrupt, one per TBU.

Name	Protocol	Type	Description
tbu_pmusnapshot_ack[62]	Signal	Master	PMU snapshot interface for the TBU, ack a snapshot.
tbu_pmusnapshot_req[62]	Signal	Slave	PMU snapshot interface for the TBU, request a snapshot.
tbu_ras_cri[62]	Signal	Master	Critical error interrupt for RAS events from the TBU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE that in the MMU-700 model this is called tbu_cri_irpt.
tbu_ras_eri[62]	Signal	Master	Error Recovery Interrupt for RAS events from the TBU. This is used for uncorrected errors. NOTE that in the MMU-700 model this is called tbu_eri_irpt. NOTE that in the MMU-700 model this is called tbu_eri_irpt.
tbu_ras_fhi[62]	Signal	Master	Fault Handling Interrupt for RAS events from the TBU. Usually this is for corrected errors. NOTE that in the MMU-700 model this is called tbu_fhi_irpt.
tbu_ras_lt_cri[62]	Signal	Master	Level triggered critical error interrupt for RAS events from the TBU.
tbu_ras_lt_eri[62]	Signal	Master	Level triggered error recovery interrupt for RAS events from the TBU.
tbu_ras_lt_fhi[62]	Signal	Master	Level triggered fault handling interrupt for RAS events from the TBU.
tbu_ras_lt_irpt_v[62]	Signal	Master	Level triggered valid output for connection to System RAS agents. Asserted when the RAS record contains at least one valid error.
tbu_reset_in[62]	Signal	Slave	Reset signals The TBUs can have independent reset signals. However, TCU reset will result in the reset of all TBUs. This behavior is unlike SMMU RTL implementations which do allow for independent reset of the TCU separate from TBUs. Each signal tbu_reset_in[n] corresponds to the TBU using tbs_pvbus_s[n]/tbs_pvbus_s[n] pair. If the SMMU receives a transaction whilst the TBU is expected to be in reset then it will complain using the ArchMsg.Warning.warning trace source. Those tbu_reset_in that correspond to a PCIe-RC connection can be connected to monitor the PCIe-RC's reset signal. If it receives an ATS request when in reset then it will complain in a similar way. You must connect these pins if you wish the TCU_NODE_STATUS for the nodes to be accurate (including any connected to the PCIe-RC).
tcu_pmu_irpt	Signal	Master	TCU Performance Monitoring Unit interrupt
tcu_pmusnapshot_ack	Signal	Master	PMU snapshot interface for the TCU, ack a snapshot.
tcu_pmusnapshot_req	Signal	Slave	PMU snapshot interface This is a four-phase handshake to have the corresponding PMCG perform a capture of the current counter values. PMU snapshot interface for the TCU, request a snapshot.
tcu_ras_cri	Signal	Master	Critical error interrupt for RAS events from the TCU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE that in the MMU-700 model this is called tcu_cri_irpt.

Name	Protocol	Type	Description
tcu_ras_eri	Signal	Master	Error Recovery Interrupt for RAS events from the TCU. This is used for uncorrected errors. NOTE that in the MMU-700 model this is called tcu_eri_irpt.
tcu_ras_fhi	Signal	Master	Fault Handling Interrupt for RAS events from the TCU. Usually this is for corrected errors. NOTE that in the MMU-700 model this is called tcu_fhi_irpt.
tcu_ras_lt_cri	Signal	Master	Level triggered critical error interrupt for RAS events from the TCU.
tcu_ras_lt_eri	Signal	Master	Level triggered error recovery interrupt for RAS events from the TCU.
tcu_ras_lt_fhi	Signal	Master	Level triggered fault handling interrupt for RAS events from the TCU.
tcu_ras_lt_irpt_v	Signal	Master	Level triggered valid output for connection to System RAS agents. Asserted when the RAS record contains at least one valid error.
tcu_reset_in	Signal	Slave	The reset signal to the TCU interface.

## Parameters for MMU\_S3

### TCUCFG\_DPT\_SUPPORT

#### Type

bool

#### Default value

true

MMU-S3 r1 Parameter Only: Enable Device Permission Table (DPT), a mechanism to enforce the association between granules of physical address space and the memory footprint of virtual machines.

**0**

Realm DPT is disabled

**1**

Realm DPT is enabled (default)

### TCUCFG\_DVM\_VAS

#### Type

uint32\_t

#### Default value

53

Virtual address size used by the system. Once set, this value is discoverable using `TCU_SYSDISC35.TCUCFG_DVM_VAS`.

In hardware, it is important to get this parameter correct as it determines the DVM message format. If this doesn't match the PEs, DVM messages are misinterpreted and any TLBI operations performed are incorrectly applied.

The model uses a representation of DVM that does not depend on the VA size and so misconfiguring this has no effect other than on the system discovery register value. Accepted Values: 49 53

### **TCUCFG\_MECID\_WIDTH**

#### **Type**

uint32\_t

#### **Default value**

16

Memory Encryption Context (MEC) is a feature introduced in MMU-S3. The MECID is a 1-16 bit identifier that, if implemented, supports Memory Encryption Contexts for the Realm programming interface. The given value indicates the number of bits in the MECID. A value of 0 will disable MEC. Accepted Values: 0 4 8 12 16

### **TCUCFG\_PARTID\_WIDTH**

#### **Type**

unsigned

#### **Default value**

9

The width of the MPAM PARTID on the bus.

The value 10 is just for MMU\_S3 r1.

See also parameter mpam\_attribute\_transform. Accepted Values: 1 6 9 10

### **TCUCFG\_XLATE\_SLOTS**

#### **Type**

uint32\_t

#### **Default value**

512

Maximum number of outstanding stalled transactions that the SMMU supports.



**Note**

TCUCFG\_XLATE\_SLOTS must be  $\geq$  TCUCFG\_PTW\_SLOTS which is currently fixed to 512.

---

Accepted Values: 512 1024 2048 4096

**all\_error\_messages\_through\_trace****Type**

bool

**Default value**

false

Some conditions in the SMMU are so strange that the software programming the SMMU has done something wrong. At this point messages are output to either `ArchMsg.Error.*` or `ArchMsg.Warning.*` or to the error stream of the simulator. Outputting to the error stream of the simulator may cause it to return with a non-zero exit status.

If you set this option to true then instead of using the error stream of the simulator it will always use a trace stream allowing the simulation to exit with a zero exit status.

**axi\_stream\_msi\_TDEST****Type**

uint32\_t

**Default value**

0

ID of the AXI stream subordinate port on the GIC that receives SMMU originated MSIs sent directly. The name of the GIC port is `axi_stream_msi_s`.

**Note**

If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TID`

**axi\_stream\_msi\_TID****Type**

uint32\_t

**Default value**

0

ID of the AXI stream manager port that sends SMMU TCU originated MSIs directly to the GIC. The name of the SMMU port is `axi_stream_msi_m`.

**Note**

If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TDEST`

**`axi_stream_msi_addr_to_match`****Type**

`uint64_t`

**Default value**

`0xFFFFFFFFFFFFFFFF`

If the last two bits are 0, any SMMU-originated MSI which exactly matches the address will be sent through the AXI stream port `axi_stream_msi_m` which is usually connected to the GIC.

This parameter drives the value of the `axi_stream_msi_addr_to_match_s` port at simulation reset. For every reset after that, the value of the port will be sampled and used if changed.

---

**Note**

The entire address must match, including bits [1:0]. As MSIs are 32 bit aligned then if `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_TID`
- `axi_stream_msi_TDEST`

**`behaviour_of_sampled_at_reset_signals`****Type**

unsigned

**Default value**

0

Some configuration signals into the SMMU are sampled on negedge of reset.

However, it can sometimes be hard to arrange to drive a configuration pin before the negedge of reset.

The configuration pins are sampled:

**0**

at negedge reset.

**1**

at negedge reset, but if a later change occurs at the same simulated time, and no transactions have occurred, then they will be resampled and the SMMU reset again.

**cmdq\_max\_number\_of\_commands\_to\_buffer****Type**

uint32\_t

**Default value**

10

The command queues can buffer fetched commands before issuing them. This parameter is roughly the maximum number of commands to do this for. The programmer visible effects are that just because the CONS pointer shows a command has been consumed does not necessarily mean that it has been issued (and completed). Higher values accentuate this effect.

**enable\_device\_id\_checks****Type**

bool

**Default value**

true

If this parameter is true then the DeviceIDs seen by the GIC are:

- **for client devices**

$$\text{DeviceID} = \text{StreamID} + \text{translated\_device\_id\_base}$$

- **for SMMU-generated MSIs**

$$\text{smmu\_msi\_device\_id}$$

This parameter enables two checks:

- If the DeviceID is used in the `output_attribute_transform` parameter, if it overflows 32 bits then the model warns. If the DeviceID is not used, it is assumed that the external agent that forms the DeviceID warns if it overflows.
- If the SMMU supports MSIs, the model checks that the GIC is able to distinguish an MSI generated by the SMMU from one generated by a client device.

As the exact mechanism to determine the DeviceID is in the system and not necessarily under control of the SMMU then you can disable these warnings using this parameter.

See also the parameters: `output_attribute_transform` and `msi_attribute_transform`.

**hide\_warning\_ACCESSEN\_GPCEN\_set\_to\_1\_in\_a\_single\_write****Type**

bool

**Default value**

false

The architecture recommends against setting `SMMU_ROOT_CR0.ACCESSEN` and `SMMU_ROOT_CR0.GPCEN` to 1 in the same write if it is possible that a concurrent client device transaction could appear at the SMMU. This is because there could be a time window where the client device transaction sees effective values `ACCESSEN == 1` and `GPCEN == 0` and so would bypass GPC checking.

If your system cannot generate this possibility then you can set this parameter to true and turn off the warning that software has set both `ACCESSEN` and `GPCEN` to 1 at the same time.

**hide\_warning\_EOPD\_differs\_from\_what\_would\_be\_cached****Type**

bool

**Default value**

false

When this parameter is set to true, warnings that the effective EOPD value differs from what would be cached in the TLB are disabled. False (warnings are showed) by default.

**hide\_warning\_NoStreamID\_transaction\_for\_unsupported\_PAS\_or\_MPAM\_SP****Type**

bool

**Default value**

false

When RME is not supported then a NoStreamID transaction with `PAS[1] == 1` or `MPAM_SP[1] == 1` is treated as though `PAS[1] == 0` and `MPAM_SP[1] == 0`. This is usually a system construction error and is not expected to occur.

The SMMU warns when this occurs, but the warning can be hidden by setting this parameter.

**howto\_identify****Type**

string

**Default value**

"use-identify"

If `use-identify` then the SMMU uses the `identify` port to determine the `ssd`, `streamID`, `substreamID`. Otherwise, this string extracts them from the transaction's attributes.



Examples:

```
SEC_SID=ExtendedID[63], SSV=ExtendedID[62], SubstreamID=ExtendedID[51:32],
StreamID=ExtendedID[31:0]
```

or

```
nSEC_SID=ExtendedID[63], StreamID=ExtendedID[55:24], nSSV=ExtendedID[20],
SubstreamID=ExtendedID[19:0]
StreamID[31:24]=0, StreamID[23:0]=ExtendedID[23:0], SSV=1[0], ...
```



If you are not using the `use-identify` option then the configuration string is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

LHS Symbols:

**SIDV**

Indicates that the StreamID is valid.

**SSV**

Indicates that the SubstreamID is valid.

**SEC\_SID / SEC\_SID\_bit\_1**

Bits 0 and 1 respectively of the StreamID Security State

**StreamID**

(32 b) valid if `SIDV` is 1 or both `SIDV` and `nSIDV` are unused.

**SubstreamID**

(20 b) is valid if `ssv` is true.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

RHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming PVBUS transaction attributes

`SEC_SID` is 1b in the model. For RME systems, in hardware, then `SEC_SID` is 2b, in the model `SEC_SID_bit_1` represents bit[1].

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

Negative and positive logic symbols for the same attribute is an error.

The model uses the term SSD (Security State Determination) to mean which security state the transaction, register, structure, event, etc. belongs to.

In the SMMUV3 Architecture, the side-band signal `SEC_SID` holds the information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one bit signal. With RME, in the hardware, it was expanded to 2b. To retain backwards compatibility in the model, `SEC_SID` remains as 1b in this parameter, but the second bit can be expressed with `SEC_SID_bit_1` (and its negative logic version `nSEC_SID_bit_1`)

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

For those systems that support NoStreamID transactions, and `howto_identify` is not using the port, first the SMMU determines the value of `SIDV` or `NSIDV` to see if the transaction is a NoStreamID transaction (`SIDV == 0` or `NSIDV == 1`). If so then the rest of the `howto_identify` string is ignored as the information extracted relates only to StreamID transactions. Instead more information is extracted using the parameter `howto_identify_NoStreamID_extra_info`.

In AMBA systems, the equivalent signal to `SIDV` is called either `ARMMUVALID` or `AWMMUVALID`. Collectively they are known as `AxMMUVALID`.

### **`howto_identify_NoStreamID_extra_info`**

#### **Type**

string

#### **Default value**

""


The behavior of this parameter depends on `howto_identify`

- if it equals 'use-identify' then this must be "", otherwise there is an error.
- if it identifies a NoStreamID transaction (`SIDV=0`) then this parameter includes one or more of
  - `MPAM_SP`
  - `MPAM_PARTID`
  - `MPAM_PMG`
  - `MECID`
  - `HWATTR_KIND_0`
- in any other case, this parameter is ignored.

Fields set in this parameter must not overlap the `SIDV/NSIDV` fields in `howto_identify`

Example:

```
MPAM_PMG[7:0]=ExtendedID[62:55], MPAM_PARTID[15:0]=ExtendedID[54:39],
MPAM_SP[1:0]=ExtendedID[38:37], MECID[15:0]=UserFlags[31:16]
HWATTR_KIND_0[3:0]=ExtendedID[42:39]
```



Note


The parameter is parsed according to the rules in the section ‘Parameters for parsing transaction attributes’.

**ish\_is\_osh\_DANGER**

Type  
bool


Default value  
true

When set, any transaction that would use the architectural inner shareable domain is converted to use the outer shareable domain.



Note

This parameter should match the equivalent `ish_is_osh` from the PE. If an incompatible value of the `ish_is_osh` parameter is configured for the PE and the SMMU, data coherency may be compromised.



Note

All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

**legacy\_tz\_en**

Type  
bool

Default value  
false

The default value of the `legacy_tz_en` pin:

- 0  
RME is enabled
- 1  
RME is disabled

**list\_of\_ns\_sid\_high\_at\_bitpos0****Type**

string

**Default value**

""

A comma-separated list of values to bitwise OR into each Non-secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems. The effective value for any PCIe-RC ports must be the same for non-secure and realm. See `list_of_r_sid_high_at_bitpos0`.

The empty string corresponds to all 0s.

**list\_of\_pcie\_mode****Type**

string

**Default value**

""

A comma-separated list of ranges of ports that represent TBUs that are attached to PCIe Root Complexes (PCIe-RC). A single PCIe-RC might use several TBUs and stripe accesses across them. The attribute handling for these TBUs is slightly different in that if the PCIe transaction is NoSnoop and the output attributes of the translation would be weaker than `INC-ONC-OSH` then the output is forced to `INC-ONC-OSH`.

`INC-ONC-OSH` == "inner normal non-cacheable, out normal non-cacheable, outer shared"

**list\_of\_pcie\_rc****Type**

string

**Default value**

""

This is a list of ports that are connected to PCIe Root Complex (PCIe-RC) by a protocol called DTI-ATS. This port is used to transport ATS and PRI Requests to the SMMU from the PCIe-RC.

In the real hardware, the PCIe-RC uses this port for ATS/PRI, and the actual transactions go through separate TBUs. In the model, this port can accept actual transactions as well. However, in the model, then the ATC Invalidates and the PRI Responses need to be transferred over the corresponding `pvbus_id_routed_m` port as DTI-ATS is bidirectional, but PVBus is not.

**list\_of\_r\_sid\_high\_at\_bitpos0****Type**

string

**Default value**

""

A comma-separated list of values to bitwise OR into each Realm StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID. This only has an effect for r1 and later.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems. The effective value for any PCIe-RC ports must be the same for non-secure and realm. See `list_of_ns_sid_high_at_bitpos0`.

The empty string corresponds to all 0s.

"use-ns" can be used to apply the `list_of_ns_sid_high_at_bitpos0` values instead.

**list\_of\_s\_sid\_high\_at\_bitpos0****Type**

string

**Default value**

""

A comma-separated list of values to bitwise OR into each Secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

The empty string corresponds to all 0s.

use-ns can be used to apply the `list_of_ns_sid_high_at_bitpos0` values instead.

**mec\_attribute\_transform****Type**

string

**Default value**

""

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MEC is supported, this is applied to *all* downstream transactions to transport the MEC information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"UserFlags[31:16]=MECID[15:0]"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/MasterID/UserFlags.

RHS Symbols:

- MECID

Any bits with no transform are unchanged.



Note

Attribute transforms applied before this:

- for client transactions `output_attribute_transform / output_attribute_transform_for_NoStreamID`.
- for table walks `tw_qs_attribute_transform`.
- for MSIs `msi_attribute_transform`.
- if MPAM is enabled `mpam_attribute_transform`.

## **mpam\_attribute\_transform**

Type

string

Default value

"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID,  
ExtendedID[38]=MPAM\_NS"



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MPAM is supported, this is applied to *all* downstream transactions to transport the MPAM information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID,  
ExtendedID[38]=MPAM_SP[0]"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/MasterID/UserFlags.

RHS Symbols:

- MPAM\_PARTID
- MPAM\_PMG
- MPAM\_NS
- MPAM\_SP
- numeric literals

Any bits with no transform are unchanged.



Note

- attribute transforms applied before this:
  - for client transactions `output_attribute_transform / output_attribute_transform_for_NoStreamID`.
  - for table walks `tw_qs_attribute_transform`.
  - for MSIs `msi_attribute_transform`.
- `mec_attribute_transform` is applied after this.
- for translated transactions from client devices then `MPAM_NS = ! SEC_SID`.

## **msi\_attribute\_transform**

### **Type**

string

### **Default value**

"ExtendedID[31:0]=smmu\_msi\_device\_id, ManagerID64[31:0]=0xFFFFffff"

Transform downstream attributes of MSI transactions.



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter output attributes of SMMU-generated MSIs. Example:

```
"UserFlags[15:0]=smmu_msi_device_id[31:16],
ManagerID64[15:0]=smmu_msi_device_id[15:0],
ExtendedID=0"
```

LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Outgoing PVBUS transaction attributes

RHS Symbols:

**smmu\_msi\_device\_id**

The value stored in the parameter with the same name

**interrupt\_kind**

The selected bit corresponds to the interrupts listed below

**0/1**

EVENTQ s/ns

**2**

PRIQ

**3/4**

CMD\_SYNC s/ns

**5/6**

GERROR s/ns

**7/8**

PMCG s/ns

**9/10/11**

RAS FHI/ERI/CRI

**12/13**

gpf\_far/gpt\_cfg\_far

**14/15/16/17**

Realm EVENTQ/PRIQ/CMDQ/GERROR

**HWATTR\_KIND\_0**

PBHA information

**Numeric Literals**

Any number. Ex: 0x1234

ExtendedID/ManagerID64/MasterID/UserFlags start with values {0, 0xFFFFffff, 0xFFFFffff, 0} respectively.

Any bits with no transform are unchanged.

This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices.



**Note**

See also `output_attribute_transform` and `enable_device_id_checks`.

**Note**

After 11.25 the `interrupt_kind` field was extended to 5 bits. This is strictly a non-backwards compatible change. However, the original 3 bits were insufficient to express all the interrupt kinds that exist.

---

## **`normalize_input_normal_non_iWB_oWB_to_iNC_oNC_osh_DANGER`**

### **Type**

bool

### **Default value**

true

When set, use Inner Non Cacheable, Outer Non Cacheable, Outer Shareable for any upstream transaction that would use any of the following attributes:

- Normal Non-cacheable Bufferable
- Normal Non-cacheable Non-bufferable
- Write-through

**Note**

This parameter should match the equivalent configuration from the PE. If an incompatible value of this parameter is configured for the PE and the SMMU, data coherency may be compromised.

All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

---

## **`ns_gbpa_abort_init`**

### **Type**

bool

### **Default value**

false

The default value of the tie off signal `ns_gbpa_abort_init`

## **`number_of_ports`**

### **Type**

unsigned

**Default value**

1

The number of port pairs that the SMMU has.

**output\_attribute\_transform****Type**

string

**Default value**

"ExtendedID[31:0]=DeviceID"

Transform downstream attributes of StreamID transactions.

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID,
ManagerID64[10]=ManagerID64[11], ManagerID64[11]=ManagerID64[10]"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**DeviceID**

StreamID + translated\_device\_id\_base

**StreamID / SubstreamID / SEC\_SID / SEC\_SID\_bit\_1 / SIDV / SSV**

Architectural information. See parameter `howto_identify` for more information.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

**St1PBHA / St2PBHA**

Page Based Hardware Attributes from leaf descriptors (zero if unused).

**STE\_IMPDEF1**

STE[127:116]

**HWATTR\_KIND\_0**

PBHA information

**Numeric Literals**

Any number. Ex: 0x1234

The streamID has had ns\_sid\_high/s\_sid\_high/r\_sid\_high ORred into it for the appropriate TBU.



Note

- mpam\_attribute\_transform and mec\_attribute\_transform are applied in order after this.
- See also output\_attribute\_transform\_for\_NoStreamID for NoStreamID transactions.

**output\_attribute\_transform\_for\_NoStreamID**

**Type**

string

**Default value**

"ExtendedID[31:0]=0, ExtendedID[32]=1"



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

Transform downstream attributes of NoStreamID transactions.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=0, UserFlags[31]=1, UserFlags[19:0]=0,  
ManagerID64[10]=ManagerID64[11],  
ManagerID64[11]=ManagerID64[10] ManagerID64[9:6]=HWATTR_KIND_0"
```

RHS/LHS Symbols: \* ExtendedID/ManagerID64/MasterID/UserFlags: incoming/outgoing attributes.

RHS Symbols: \* SIDV = 0, nSIDV = 1 (fixed values to indicate NoStreamID) \* PAS \* HWATTR\_KIND\_0

Any bits with no transform are unchanged.



Note

- mpam\_attribute\_transform and mec\_attribute\_transform are applied in order after this.
- see also output\_attribute\_transform for StreamID transactions.

**output\_id\_routed\_transform****Type**

string

**Default value**

"Address[27:12]=StreamID[15:0], PAS=SSD"



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

The SMMU generates the following ID-routed transaction on the pvbus\_id\_routed\_m bus:

- ATC Invalidate
- PRI Response

This parameter controls how the SMMU should express:

- the StreamID
- the Trusted (T) bit

The value is a comma-separated list of assignments:

```
Address[27:12]=StreamID[15:0], ExtendedID[60]=T, ExtendedID[15:0]=StreamID[31:16]
```

Address bits[11:0] cannot be used.

The LHS can be one of:

- PAS
- MasterID / ManagerID64 / ExtendedID / UserFlags
- Address

The RHS can be one of:

- a numeric constant
- SSD
- T or negative version nT
- StreamID

For realm (or 'Trusted') transactions, then `ssd=0b11`, `T=1`, `nT=0`. For non-secure (or 'Non-Trusted') transactions, then `ssd=0b01`, `T=0`, `nT=1`.

**prefetch\_only\_requests****Type**

unsigned

**Default value**

0

The simulator supports 'prefetch-only' DMI requests, which can occur at any time and for any reason and are intended to be invisible to the end execution of the model and to the user.

**0**

deny all prefetch-only requests

**1**

- use debug requests for any page table walks
  - form and use debug TLB/cache entries
  - any faults will not record, but deny the prefetch request

**2**

- treat prefetch-only requests like normal transactions
  - use normal page table walk transactions
  - use and form normal TLB/cache entries
  - faults will alter the programmer-visible state of the SMMU

0 is the safest.

1 treats the access like a debug request and requires that debug page table walks are treated correctly downstream. Any descriptors that need HTTU to allow the transaction to proceed will fail the request.

2 is dangerous, it uses real transactions and reports faults that are unphysical. Real transactions can be `wait()`ed and this disobeys the SystemC spec for `get_direct_mem_ptr()`.

**rme\_logpt\_entry\_covers\_log2size\_in\_bytes****Type**

uint32\_t

**Default value**

30

Each LOGPT entry covers:

```
2**rme_logpt_entry_covers_log2size_in_bytes
```

bytes of address space.

The valid values for this parameter are: 30, 34, 36, 39

This parameter is reported in an encoded format as the read-only field:

```
SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ
```

This parameter can be overridden by the port `logptsz_s` when sampled on negedge of reset.

### **s\_gbpa\_abort\_init**

#### **Type**

bool

#### **Default value**

false

The default value of the tie off signal `s_gbpa_abort_init`

### **sec\_override**

#### **Type**

bool

#### **Default value**

false

The IMP DEF port `sec_override` controls whether some of the registers are accessible to secure or non-secure/realm transactions. This parameter is the default value assumed for that port if the port is not driven by a signal.

### **seed**

#### **Type**

uint32\_t

#### **Default value**

0x12345678

Used to seed the pseudo-random number generator that the SMMU model uses.

### **size\_of\_cd\_cache**

#### **Type**

uint32\_t

#### **Default value**

0

The number of entries in the cache holding CD structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_dpttlb****Type**

uint32\_t

**Default value**

0

The number of entries in the DPT TLB. If this is zero then it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_l1cd\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding L1CD descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_l1ste\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding L1STE descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_ste\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding STE structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_tlb****Type**

uint32\_t

**Default value**

0

The number of entries in the TLB. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**smmu\_msi\_device\_id****Type**

uint32\_t

**Default value**

0

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID.

See parameters `msi_attribute_transform` and `enable_device_id_checks`.

**sup\_btm****Type**

bool

**Default value**

true

The default value of the register field `SMMU_IDR0.BTM` and `SMMU_ROOT_IDR0.BGPTM` (when supported). This enables Broadcast TLB maintenance.

**sup\_cohacc****Type**

bool

**Default value**

true

The default value of the register `SMMU_IDR0.COACC`

**sup\_httu****Type**

bool

**Default value**

true

The initial value of the `sup_httu` port. See the port description for `sup_httu`.

**sup\_oas****Type**

unsigned



**Default value**

6

The hardware has an input port `sup_oas[2:0]` that indicates what output address size (OAS) the system has. This is sampled at reset.

The model does not have this port as it is expected to be a constant for the system and not to change. Instead it is just a parameter.

The allowed values are:

**0**

32 bits

**1**

36 bits

**2**

40 bits

**3**

42 bits

**4**

44 bits

**5**

48 bits

**6**

52 bits

**sup\_sev****Type**

bool

**Default value**

true

The default value of the register `SMMU_IDR0.SEV`

**tlb\_when\_do\_f\_tlb\_conflict\_on\_overlap****Type**

unsigned

**Default value**

0

If a TLB entry is created by a walk and it overlaps an existing entry, there are some architectural situations where the result is known. For all others, then an implementation is allowed to use an **UNPREDICTABLE** combination of the two entries, or it can generate `F_TLB_CONFLICT`:

**0**

never generate

**1**

sometimes generate

**2**

always generate

Conflicts between global and non-global entries are not detected by the model.

**translated\_device\_id\_base****Type**

uint32\_t

**Default value**

0

When appropriately enabled, assume that client device accesses are translated to a DeviceID as seen by the GIC of:

```
StreamID + translated_device_id_base
```

See parameter `output_attribute_transform` and `enable_device_id_checks`.

**tw\_qs\_attribute\_transform****Type**

string

**Default value**

""

Transform downstream attributes of table walk and queue transactions.

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**HWATTR\_KIND\_0**

PBHA information

**kind**

Transaction kind

- For a read:
  - 0/1**  
L1STE/STE
  - 2/3**  
L1CD/CD
  - 4/5**  
S1/S2 TTD (including CAS)
  - 6**  
CMDQ
  - 7**  
VMS
  - 11/12**  
LOGPT/L1GPT
  - 13/14**  
LODPT/L1DPT
- For a write:
  - 0**  
EVENTQ
  - 1**  
PRIQ

ExtendedID/ManagerID64/MasterID/UserFlags start with values {0, 0xFFFFFFFF, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.



See also `output_attribute_transform` and `msi_attribute_transform`.

---

**version****Type**

string

**Default value**

"rOp0"

The version of this product.

Valid values are:

- rOp0
- r1p0

**wait\_cmdq\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing something on the command queue. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffffff)) - 1]$ .

**wait\_eventq\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing something on the event queue. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffffff)) - 1]$ .

**wait\_misc\_async\_actions\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing an async action that could be delayed is performed. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffffff)) - 1]$ .

**wait\_msi\_ticks**

Type  
uint64\_t

Default value  
0

This is the time to wait before sending an MSI. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFfff)) - 1]$ .

**wait\_pri\_req\_ticks**

Type  
uint64\_t

Default value  
0

This is the time to wait before processing a PRI Request. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFfff)) - 1]$ .

**wait\_pri\_resp\_ticks**

Type  
uint64\_t

Default value  
1

This is the time to wait before sending a PRI Response back to the PCIe subsystem. When a PRI Response is an auto-response then the ATC might immediately make a new ATS request, that immediately fails, immediately makes a PRI Request, or auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFfff)) - 1]$ .

3.10.66 MemoryMappedCounterModule

Memory Mapped Counter Module for Generic Timers. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1113: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Changes in 11.29.19

Parameters removed:

- `non_arch_fixed_frequency`
- `use_real_time`

## About MemoryMappedCounterModule

This component must be used by multicluster models. It also must be used to run a single core system with a timer that runs at a rate that is different to the input clock to the core.



The component has two bus slave ports because the architecture specification permits you to map each set of registers at different, non-contiguous base addresses.

## Iris and MTI instances for MemoryMappedCounterModule

This model has the following Iris instances:

**Table 3-1114: MemoryMappedCounterModule Iris instances**

InstanceName	ComponentName
MemoryMappedCounterModule	MemoryMappedCounterModule
MemoryMappedCounterModule.pvbus_control_s[0]	PVBusSlave
MemoryMappedCounterModule.pvbus_read_s[0]	PVBusSlave

This model has the following MTI trace components:

**Table 3-1115: MemoryMappedCounterModule MTI instances**

InstanceName	ComponentName
MemoryMappedCounterModule.pvbus_control_s[0]	PVBusSlave
MemoryMappedCounterModule.pvbus_read_s[0]	PVBusSlave

## Ports for MemoryMappedCounterModule

**Table 3-1116: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	This clock input determines the frequency of the Physical Count provided to the clusters connected to the cntvalueb port.
cntvalueb	<a href="#">CounterInterface</a>	Master	This master port implements a private protocol between the cluster and the MemoryMappedCounterModule. This must be connected to the cntvalueb port on each cluster in the system and to the MemoryMappedCounterModule component.
counter_reset	<a href="#">Signal</a>	Slave	Resets when set.
pvbus_control_s	<a href="#">PVBus</a>	Slave	This slave port provides memory-mapped read write access to the control registers of the module.
pvbus_read_s	<a href="#">PVBus</a>	Slave	This slave port provides memory-mapped read access to the status frame registers.

## Parameters for MemoryMappedCounterModule

### **base\_frequency**

#### Type

int

#### Default value

0x5f5e100

#### Description

Reset value for CNTFID0, base frequency in Hz.

### **cntcidr0123\_C**

#### Type

int

#### Default value

0x0

#### Description

Values to be returned for control-frame CIDR registers.

### **cntcidr0123\_R**

#### Type

int

#### Default value

0x0

#### Description

Values to be returned for read-frame CIDR registers.

### **cntpidr0123\_C**

#### Type

int

#### Default value

0x0

#### Description

Values to be returned for control-frame PIDR registers 0-3.

### **cntpidr0123\_R**

#### Type

int

#### Default value

0x0

**Description**

Values to be returned for read-frame PIDR registers 0-3.

**cntpidr4567\_C****Type**

int

**Default value**

0x0

**Description**

Values to be returned for control-frame PIDR registers 4-7.

**cntpidr4567\_R****Type**

int

**Default value**

0x0

**Description**

Values to be returned for read-frame PIDR registers 4-7.

**diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

**has\_additional\_registers****Type**

bool

**Default value**

0x0

**Description**

Implements additional REFCLK CNT control registers.

**has\_counter\_scaling****Type**

bool

**Default value**

0x0



**Description**

Implements ARMv8.4 generic counter scaling (FEAT\_CNTSC).

**non\_arch\_start\_at\_default**

**Type**

bool

**Default value**

0x0

**Description**

Firmware is expected to enable the timer at boot time. However, turning this parameter on is a model-specific way of enabling the counter module out of reset.

**readonly\_is\_WI**

**Type**

bool

**Default value**

0x0

**Description**

Ignore (rather than failing) on writes to read-frame.

**3.10.67 MessageHandlingUnit**

Message Handling Unit. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1117: IP revisions support**

Revision	Quality level
v2.0	Full support
v2.1	Full support
v3	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Iris and MTI instances for MessageHandlingUnit**

This model has the following Iris instances:

**Table 3-1118: MessageHandlingUnit Iris instances**

InstanceName	ComponentName
MessageHandlingUnit	MessageHandlingUnit
MessageHandlingUnit.a_to_b_v2	MessageHandlingUnitV2
MessageHandlingUnit.a_to_b_v3	MessageHandlingUnitV3

InstanceName	ComponentName
MessageHandlingUnit.version_mapper_a_to_b_rec	PVBusMapper
MessageHandlingUnit.version_mapper_a_to_b_snd	PVBusMapper

This model has the following MTI trace components:

**Table 3-1119: MessageHandlingUnit MTI instances**

InstanceName	ComponentName
MessageHandlingUnit.a_to_b_v2	MessageHandlingUnitV2
MessageHandlingUnit.a_to_b_v3	MessageHandlingUnitV3
MessageHandlingUnit.version_mapper_a_to_b_rec	PVBusMapper
MessageHandlingUnit.version_mapper_a_to_b_snd	PVBusMapper

## Ports for MessageHandlingUnit

**Table 3-1120: Ports**

Name	Protocol	Type	Description
pdbus_s_rec	PVBus	Slave	-
pdbus_s_snd	PVBus	Slave	-
rec_combined_channel_irq_out[200]	Signal	Master	-
rec_combined_irq_out	Signal	Master	-
rec_reset_in	Signal	Slave	-
snd_combined_channel_irq_out[200]	Signal	Master	-
snd_combined_irq_out	Signal	Master	-
snd_reset_in	Signal	Slave	-

## Parameters for MessageHandlingUnit

### NUM\_DB\_CH

#### Type

int

#### Default value

0x1

#### Description

Number of doorbell channels.

### NUM\_FAST\_CH

#### Type

int

#### Default value

0x1

**Description**

Number of fast channels.

**a\_to\_b\_v3.NUM\_FIFO\_CH**

**Type**

int

**Default value**

0x1

**Description**

Number of FIFO Channels, default=1.

**a\_to\_b\_v3.auto\_op\_full**

**Type**

bool

**Default value**

0x0

**Description**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

**a\_to\_b\_v3.fast\_ch\_group\_int\_enable**

**Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**a\_to\_b\_v3.fifo\_depth**

**Type**

int

**Default value**

0x4

**Description**

Depth of the FIFO = fifo\_depth + 1.

**a\_to\_b\_v3.m16ba\_spt**

**Type**

bool

**Default value**

0x0

**Description**

Mailbox 16 bit access support to FIFO registers.

**a\_to\_b\_v3.m32ba\_spt**

**Type**

bool

**Default value**

0x1

**Description**

Mailbox 32 bit access support to FIFO registers.

**a\_to\_b\_v3.m64ba\_spt**

**Type**

bool

**Default value**

0x0

**Description**

Mailbox 64 bit access support to FIFO registers.

**a\_to\_b\_v3.m8ba\_spt**

**Type**

bool

**Default value**

0x0

**Description**

Mailbox 8 bit access support to FIFO registers.

**a\_to\_b\_v3.monolithic**

**Type**

bool

**Default value**

0x1

**Description**

Monolithic or Distributed MHU - default: monolithic(true).

**a\_to\_b\_v3.p16ba\_spt**

**Type**

bool

**Default value**

0x0

**Description**

Postbox 16 bit access support to FIFO registers.

**a\_to\_b\_v3.p32ba\_spt**

**Type**

bool

**Default value**

0x1

**Description**

Postbox 32 bit access support to FIFO registers.

**a\_to\_b\_v3.p64ba\_spt**

**Type**

bool

**Default value**

0x0

**Description**

Postbox 64 bit access support to FIFO registers.

**a\_to\_b\_v3.p8ba\_spt**

**Type**

bool

**Default value**

0x0

**Description**

Postbox 8 bit access support to FIFO registers.

**diagnostics**

**Type**

int

**Default value**

0x2

**Description**

Diagnostics 0==FATAL\_ERROR -> 4==DEBUG.

**fast\_ch\_group\_int\_enable**

**Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**fast\_ch\_n\_per\_group****Type**

int

**Default value**

0x1

**Description**

Fast Channel num channels per group, default=1.

**fast\_ch\_num\_groups****Type**

int

**Default value**

0x1

**Description**

Fast Channel num of groups, default=1.

**fast\_ch\_word\_size****Type**

int

**Default value**

0x20

**Description**

Fast Channel word size 32bit or 64bit, default=32.

**major\_version****Type**

int

**Default value**

0x2

**Description**

MHU major version (default=2).

**mhu\_arch\_beta01****Type**

bool

**Default value**

0x0

**Description**  
true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

**minor\_version**

**Type**  
int  
**Default value**  
0x1

**Description**  
MHU minor version (default=1).

**product\_id**

**Type**  
int  
**Default value**  
0x0

**Description**  
MHU part number.

3.10.68 MessageHandlingUnitV2

Message Handling Unit Version 2. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1121: IP revisions support

Revision	Quality level
N/A	Full support

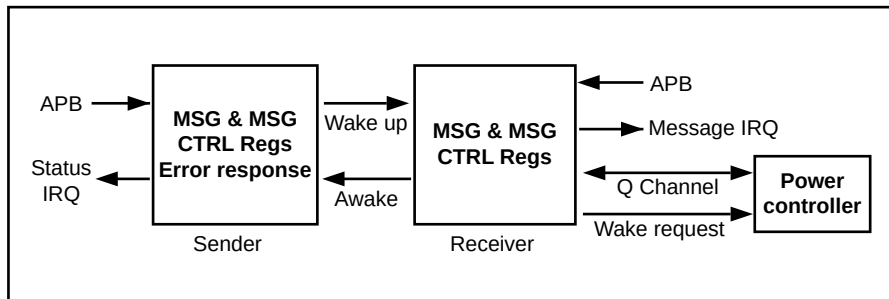
For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About MessageHandlingUnitV2

MessageHandlingUnitV2 is a unidirectional message channel with two APB interfaces, one for the sender and one for the receiver. There is one message handling unit for each sender and receiver pair.

The receiver can be powered off. A Q-Channel is provided for power control on the receiver side.

The message payload can be written directly to status registers. Multiple channels can be combined into a single message.

**Figure 3-3: MessageHandlingUnitV2 structure**

## Iris and MTI instances for MessageHandlingUnitV2

This model has the following Iris instances:

**Table 3-1122: MessageHandlingUnitV2 Iris instances**

InstanceName	ComponentName
MessageHandlingUnitV2	MessageHandlingUnitV2

This model has the following MTI trace components:

**Table 3-1123: MessageHandlingUnitV2 MTI instances**

InstanceName	ComponentName
MessageHandlingUnitV2	MessageHandlingUnitV2

## Ports for MessageHandlingUnitV2

**Table 3-1124: Ports**

Name	Protocol	Type	Description
int_access_nr2r	Signal	Master	-
int_access_r2nr	Signal	Master	-
mhu_combined_irq	Signal	Master	-
mhu_irq[124]	Signal	Master	-
mhu_snd_irq[124]	Signal	Master	-
pvbus_s_rec	PVBus	Slave	-
pvbus_s_snd	PVBus	Slave	-
qchannel_mhu_pwr	PChannel	Slave	-
reset_rec	Signal	Slave	-
reset_snd	Signal	Slave	-
snd_combined_irq	Signal	Master	-
wakerequest	Signal	Master	-



Parameters for MessageHandlingUnitV2

NUM\_CH

Type

int

Default value

0x1

Description

Number of device channels.

minor\_revision

Type

int

Default value

0x0

Description

MHUV2 minor revision, 0 for v2.0, 1 for v2.1.

product\_id

Type

int

Default value

0x0

Description

MHUV2 Product ID, MHU Part Number.

3.10.69 MessageHandlingUnitV3

Message Handling Unit Version 3. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1125: IP revisions support

Revision	Quality level
v3	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Iris and MTI instances for MessageHandlingUnitV3

This model has the following Iris instances:

**Table 3-1126: MessageHandlingUnitV3 Iris instances**

InstanceName	ComponentName
MessageHandlingUnitV3	MessageHandlingUnitV3

This model has the following MTI trace components:

**Table 3-1127: MessageHandlingUnitV3 MTI instances**

InstanceName	ComponentName
MessageHandlingUnitV3	MessageHandlingUnitV3

### Ports for MessageHandlingUnitV3

**Table 3-1128: Ports**

Name	Protocol	Type	Description
pvbus_s_rec	PVBus	Slave	Register access for Receiver/Mailbox
pvbus_s_snd	PVBus	Slave	Register access for Sender/Postbox
rec_combined_irq_out	Signal	Master	All interrupts combined for Receiver/MBX
rec_fast_channel_group_irq_out[32]	Signal	Master	Receiver fast channel group interrupts
rec_fast_channel_irq_out[1024]	Signal	Master	Receiver fast channel interrupts
rec_reset_in	Signal	Slave	Reset signal for Receiver/Mailbox
snd_combined_irq_out	Signal	Master	All Interrupts combined for Sender/PBX
snd_reset_in	Signal	Slave	Reset signal for Sender/Postbox

### Parameters for MessageHandlingUnitV3

#### NUM\_DB\_CH

##### Type

int

##### Default value

0x1

##### Description

Number of DoorBell Channels, default=1.

#### NUM\_FAST\_CH

##### Type

int

##### Default value

0x1

##### Description

Number of Fast Channels, default=1.

**NUM\_FIFO\_CH****Type**

int

**Default value**

0x1

**Description**

Number of FIFO Channels, default=1.

**auto\_op\_full****Type**

bool

**Default value**

0x0

**Description**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

**diagnostics****Type**

int

**Default value**

0x2

**Description**

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG), default=2.

**fast\_ch\_group\_int\_enable****Type**

bool

**Default value**

0x0

**Description**

Fast Channel group interrupts enable, default=false.

**fast\_ch\_n\_per\_group****Type**

int

**Default value**

0x1

**Description**

Fast Channel num channels per group, default=1.

**fast\_ch\_num\_groups****Type**

int

**Default value**

0x1

**Description**

Fast Channel num of groups, default=1.

**fast\_ch\_word\_size****Type**

int

**Default value**

0x20

**Description**

Fast Channel word size 32bit or 64bit, default=32.

**fifo\_depth****Type**

int

**Default value**

0x4

**Description**

Depth of the FIFO = fifo\_depth + 1.

**m16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 16 bit access support to FIFO registers.

**m32ba\_spt****Type**

bool

**Default value**

0x1

**Description**

Mailbox 32 bit access support to FIFO registers.

**m64ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 64 bit access support to FIFO registers.

**m8ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Mailbox 8 bit access support to FIFO registers.

**mhu\_arch\_beta01****Type**

bool

**Default value**

0x0

**Description**

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

**monolithic****Type**

bool

**Default value**

0x1

**Description**

Monolithic or Distributed MHU - default: monolithic(true).

**p16ba\_spt****Type**

bool

**Default value**

0x0

**Description**

Postbox 16 bit access support to FIFO registers.

**p32ba\_spt**

Type

bool

Default value

0x1

Description

Postbox 32 bit access support to FIFO registers.

**p64ba\_spt**

Type

bool

Default value

0x0

Description

Postbox 64 bit access support to FIFO registers.

**p8ba\_spt**

Type

bool

Default value

0x0

Description

Postbox 8 bit access support to FIFO registers.

3.10.70 NI700

NI700 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1129: IP revisions support

Revision	Quality level
r0p0	Preliminary support
r1p0	Preliminary support
r2p0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Configuring and using the model

- Major IP revisions (rX) are modeled and are controlled by the `revision` parameter.

- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of NI700 components. Set it to the name of the yaml configuration file emitted by Socrates. You must use version r1p5-03rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.
- The mapping between the port number for ASNI, AMNI, HSNi, HMNI, and PMNI ports and the xxNI name used in the yml file is its index when all xxNIs of that same type are sorted in ascending alphabetical order. For example, if the yml file has three ASNIs `asni_s100_scp`, `asni_s101_dap`, and `asni_s204_periph0`:
  - `asni_s100_scp` is mapped to `pvbuss_s_asni[0]`
  - `asni_s101_dap` is mapped to `pvbuss_s_asni[1]`
  - `asni_s204_periph0` is mapped to `pvbuss_s_asni[2]`

Similarly:

- AMNIs are mapped to `pvbuss_m_amni`
- HSNIs are mapped to `pvbuss_s_hsn`
- HMNIs are mapped to `pvbuss_m_hmni`
- PMNIs are mapped to `pvbuss_m_pmni`

Additionally, NI700's parser prints the name-to-index mappings when the component parameter `print_parser_log=true`.



Note

The PMNI RTL supports up to 16 external interfaces, however the model uses a single PMNI port. Through the single PMNI port, the models supports targeting each of the 16 interfaces independently in the SAM.

The following functionality is expected to work:

- The discovery feature to determine the system address of all nodes.
- Hashed and non-hashed memory regions. They are parsed from the `mesh_config_file`.
- MPAM support. Software must configure MPAM override in ASNI nodes by enabling and configuring the `ASNI_AR_MPAM_OVERRIDE` (0x0E0) and `ASNI_AW_MPAM_OVERRIDE` (0x0E4) registers. The `GT_MPAM_SUPPORT` signal is ignored. Software must configure MPAM support in ASNI nodes by enabling and configuring the Request MPAM Override (0x0E0) register.
- IDM support. The IDM features Access control and Reset control are modeled. Starting in r1p0, non-secure versions of the `ACCESS_STATUS` and `RESET_STATUS` registers are present. The DeviceID and the information whether an xxNI has IDM enabled are parsed from the `mesh_config_file`. When an xxNI is isolated with IDM Access Control or under reset with IDM Reset, all transactions to and from that xxNI are aborted. With respect to IDM reset support, IDM reset signals are modeled and they should be connected to the managed devices that are

connected to the respective xxNI port. The register `IDM_RESET_CONTROL` is supported. The target xxNI always enters or exits IDM reset immediately and drives the reset signals accordingly. In register `IDM_RESET_STATUS`, the bitfields `active_write` and `active_read` read always zero. In registers `IDM_RESET_READID` and `IDM_RESET_WRITEID`, the bitfields `vmaster_id` and `master_id` read always zero.

- There are no software functional differences for r2p1 and r2p0 can be used in its place.

## Model limitations



Issues listed in this section have identifiers of the form [SDDKW-x]. These are for Arm internal references only.

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
  - Power, clock, and interrupt signals are not supported.
- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- The maximum number of manager Network Interfaces is 127. The maximum tested is 127 AMNIs, 127 PMNIs, and 9 HMNIs.
- The maximum number of subordinate Network Interfaces is 128. The maximum tested is 128 ASNIs and 9 HSNIs.
- The maximum voltage, power, and clock domains of 32 each have not been tested.
- There is no support for 1 stripe target in a group, additional granularities, or the additional stripe group remap functionality described in r2p0 TRM section 2.4.5.
- Remapping features not supported:
  - Tested 5 out of the maximum of 8 remap states.
  - The priority for multiple address remapping states.
  - One target remapped to a different target.
  - A single target remapped to a stripe group.
  - One stripe group remapped to a different stripe group.
  - A stripe group remapped to a single target.
- Stripe features not supported:
  - ASNI striping to an HMNI target has not been tested.
  - Limited testing of stripe groups with different numbers of targets and granularities.
  - Single target stripe.



- The hashed memory regions support is limited by Fast Models DMI. Due to the 4KB memory pages in DMI, granularities smaller than 4KB are not accounted for by the model. Thus, subsequent accesses within a 4KB address range are delivered to the same destination node.
- AMNI nodes in the model are interface-indifferent and registers do not reflect the protocol version.
- r2 CMO Response control is not supported.
- There is no revision string for r2p1. r2p0 is functionally equivalent.
- Hashing of stripe groups is limited to a granularity of 4096B.
- xSNI access to CFGNI is not limited by router connectivity defined in the `mesh_config_file`. It considers only whether the xSNI has the CFGNI target defined in its memory map.
- A reset after model startup does not reset the registers or address remap selections.
- `*_IDM_RESET_STRAP` and its effect on the endpoint soft reset and `IDM_RESET_CONTROL` register is not supported.
- IDM for power domains is not supported.
- No register visibility support for a debugger.
- No software control for the CMO terminate response in register `AMNI_CONFIG_CTL`.

### Iris and MTI instances for NI700

This model has the following Iris instances:

**Table 3-1130: NI700 Iris instances**

InstanceName	ComponentName
NI700	NI700
NI700.decoder	PVBusMapper

This model has the following MTI trace components:

**Table 3-1131: NI700 MTI instances**

InstanceName	ComponentName
NI700	NI700
NI700.decoder	PVBusMapper

### Ports for NI700

**Table 3-1132: Ports**

Name	Protocol	Type	Description
idm_reset_signal_amni[127]	Signal	Master	IDM reset signals to AMNIs.
idm_reset_signal_asni[128]	Signal	Master	IDM reset signals to ASNIs.
idm_reset_signal_hmni[127]	Signal	Master	IDM reset signals to HMNIs.
idm_reset_signal_hsni[128]	Signal	Master	IDM reset signals to HSNIs.
idm_reset_signal_pmni[127]	Signal	Master	IDM reset signals to PMNIs.
pdbus_m_amni[127]	PVBus	Master	AMNI downstream ports.

Name	Protocol	Type	Description
pvbuse_m_hmni[127]	PVBus	Master	HMNI downstream ports.
pvbuse_m_pmni[127]	PVBus	Master	PMNI downstream ports.
pvbuse_s_asni[128]	PVBus	Slave	ASNI upstream ports.
pvbuse_s_hsni[128]	PVBus	Slave	HSNI upstream ports.
reset_in	Signal	Slave	Reset signal.

Parameters for NI700

mesh\_config\_file

Type

string

Default value

""

Description

Name of a file containing mesh placement of NI700 components.

mpam\_attributes

Type

string

Default value

""

User-defined transform to be applied to bus attributes like MasterID, ExtendedID Or UserFlags, for MPAM Attributes encoded into bus attributes.

For example:

```
ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS
```

An empty string disables MPAM support.

periphbase

Type

int

Default value

0xffffffffffffffff

Description

Value for PERIPHBASE.

**print\_config**

**Type**  
bool

**Default value**  
0x0

**Description**  
Enables printing the config register addresses.

**print\_parser\_log**

**Type**  
bool

**Default value**  
0x0

**Description**  
Enables printing the yaml config parser log messages.

**revision**

**Type**  
string

**Default value**  
"r2p0"

**Description**  
Component revision. Currently supports r0p0, r1p0, r2p0.

**show\_banner**

**Type**  
uint64\_t

**Default value**  
2

Show component banner:

**0**  
supress entire banner

**1**  
suppress config file

**2+**  
show full banner

### 3.10.71 NI710AE

NI710AE Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1133: IP revisions support**

Revision	Quality level
r0p1	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About NI710AE

To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the network interfaces present as well as their configuration options. Set it to the name of the YAML configuration file emitted by Socrates. You must use the r1p7-05 version of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.

The following functionality is supported in this release:

- All NI700 features plus the APU

Programmer's view related support:

- Model supports discovery to determine the system address of the following registers:
  - Global, Voltage, Power, and Clock Domain registers.
  - ASNI, HSNI, AMNI, HMNI, PMNI, and PMU registers.
  - APU subfeature registers.

Address map related support:

- Model supports Static Maps
- Model supports remap states
- NI700 limitations that relate to static maps and remap states also apply to NI710AE

FMU related support:

- A dedicated APB port to access the FMU registers is present in the model.
- The reset values for any configuration dependent FMU registers are currently incorrect.
- FMU related behavior has not been modeled. Any writes to FMU registers are currently ignored.

## Model limitations



Note

Issues listed in this section have identifiers of the form [SDDKW-x]. These are for Arm internal references only.

- All [NI700](#) limitations apply to NI710AE.
- FMU registers are accessible via the dedicated APB port but the functionality is not modeled.
- The PMU cannot be configured.
- [SDDKW-84456] Secure Access Override functionality (`xxNI.secure_access` register) is not supported.
- IDM limitations:
  - [SDDKW-88881] IDM Functionality is not tested and may not function correctly. Contact Arm Technical Support for support interest.
  - [SDDKW-88881] IDM Registers are accessible but `IDM_DEVICE_ID` is 0 for all IDMs.
- APU limitations
  - The base and size of address regions must be aligned to 4KiB even when `apuRegion4k` is false. A warning will be given and the region may not function properly due to PVBUS limitations.
  - `APU_ENABLE_RESET_STRAP` is not supported and `APU_CTRL.apu_enable` bit should be used to enable APU instead.
  - [SDDKW-80598] APU Region Locking is currently not supported.
  - APU Model does not enforce programming order described in TRM 102756\_0001\_03 Section 3.4.4.5 “Order of programming for APU address region registers”. The APU Model creates regions from the current values of the APU registers when `APU_CTRL.apu_enable = 1` is written.
  - [SDDKW-87737] APU Reprogramming in the model does not enforce the requirement that `APU_CTRL.apu_enable = 0` is written first before `APU_CTRL.apu_enable = 1` is written to reprogram new regions.
  - [SDDKW-82664] APUID is not read from the transaction attributes, instead the model parameter `apu_subsystem_id` is used to specify the `subsystemID/APUID` of the requestor sending in a transaction through an xSNI.

### Iris and MTI instances for NI710AE

This model has the following Iris instances:

**Table 3-1134: NI710AE Iris instances**

InstanceName	ComponentName
NI710AE	NI710AE
NI710AE.decoder	PVBusMapper

This model has the following MTI trace components:

**Table 3-1135: NI710AE MTI instances**

InstanceName	ComponentName
NI710AE	NI710AE
NI710AE.decoder	PVBusMapper

## Ports for NI710AE

**Table 3-1136: Ports**

Name	Protocol	Type	Description
idm_reset_signal_amni[127]	Signal	Master	IDM reset signals to AMNIs.
idm_reset_signal_asni[128]	Signal	Master	IDM reset signals to ASNIs.
idm_reset_signal_hmni[127]	Signal	Master	IDM reset signals to HMNIs.
idm_reset_signal_hsni[128]	Signal	Master	IDM reset signals to HSNIs.
idm_reset_signal_pmni[127]	Signal	Master	IDM reset signals to PMNIs.
pvbus_m_amni[127]	PVBus	Master	AMNI downstream ports.
pvbus_m_hmni[127]	PVBus	Master	HMNI downstream ports.
pvbus_m_pmni[127]	PVBus	Master	PMNI downstream ports.
pvbus_s_asni[128]	PVBus	Slave	ASNI upstream ports.
pvbus_s_fmu_apb	PVBus	Slave	Dedicated APB port to access FMU registers.
pvbus_s_hsni[128]	PVBus	Slave	HSNI upstream ports.
reset_in	Signal	Slave	Reset signal.

## Parameters for NI710AE

### apu\_subsystem\_id

#### Type

string

#### Default value

0

APUID/SubsystemID of the component connected to each <x>SNI. Specify the subsystem id of the component connected to each <x>SNI by using a format like:

```
<x>SNI<m>=<subsystemID0>,<y>SNI<n>=<subsystemID1>
```

The subsystemID is assumed to be 0 for any component connected to an <x>SNI that does not appear in this list.

### mesh\_config\_file

#### Type

string

**Default value**

""

**Description**

Name of a file containing mesh placement of NI710AE components.

**mpam\_attributes****Type**

string

**Default value**

""

User-defined transform to be applied to bus attributes like `MasterID`, `ExtendedID` OR `UserFlags`, for MPAM Attributes encoded into bus attributes.

For example:

```
ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS
```

An empty string disables MPAM support.

**periphbase****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Value for PERIPHBASE.

**print\_config****Type**

bool

**Default value**

0x0

**Description**

Enables printing the config register addresses.

**print\_parser\_log****Type**

bool

**Default value**

0x0

**Description**

Enables printing the yaml config parser log messages.

**revision**

**Type**

string

**Default value**

"rOp1"

**Description**

Component revision. Currently supports rOp1.

**show\_banner**

**Type**

uint64\_t

**Default value**

2

Show component banner:

**0**

supress entire banner

**1**

suppress config file

**2+**

show full banner

3.10.72 NOC\_S3

NOC\_S3 Interconnect Fast Model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1137: IP revisions support

Revision	Quality level
rOp0	Preliminary support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Configuring and using the model

- Major IP revisions (rX) are modeled and are controlled by the model `revision` parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `peripheral_id2` register.



- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the network interfaces present as well as their configuration options. Set it to the name of the YAML configuration file emitted by Socrates. You must use version r1p7-06 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.
- The mapping between the port number for ASNI, AMNI, HSNI, HMNI, and PMNI ports and the xxNI name used in the yml file is its index when all xxNIs of that same type are sorted in ascending alphabetical order. For example, if the yml file has three ASNIs `asni_s100_scp`, `asni_s101_dap`, and `asni_s204_periph0`:
  - `asni_s100_scp` is mapped to `pvbuss_asni[0]`
  - `asni_s101_dap` is mapped to `pvbuss_asni[1]`
  - `asni_s204_periph0` is mapped to `pvbuss_asni[2]`

Similarly:

- AMNIs are mapped to `pvbuss_m_amni`
- HSNIs are mapped to `pvbuss_s_hsni`
- HMNIs are mapped to `pvbuss_m_hmni`
- PMNIs are mapped to `pvbuss_m_pmni`

Additionally, the mesh config file parser prints the name-to-index mappings when the component parameter `print_parser_log=true`.



Note

The PMNI RTL supports up to 16 external interfaces, however the model uses a single PMNI port. Through the single PMNI port, the models supports targeting each of the 16 interfaces independently in the SAM.

---

rXpY support:

- rOp0

Address map related support:

- Model supports Static Maps and Programmable Address Maps (PAM).
  - Static Map hashed and non-hashed memory regions are parsed from the `mesh_config_file`.
- (PAM flow) Support for `default_tgt_id` strap to set default `xsni` targets.
- Supports configurable address mask (`cmp_addr_mask_{l,u}`, `htg_addr_mask_{l,u}`).
- Model supports topology parameter `sam/regionCompLSB` to set minimum SAM address granule with values between 4KB and 64KB.
- Model supports `No_Target` as a target in Address Maps.
- Model supports remap states in static map flow.

Programmer's view related support:

- Model supports 4KB and 64KB for topology parameter `configNodeGranularity`.
- Model supports discovery to determine the system address of the following node registers:
  - Global, Voltage, Power, and Clock Domain registers.
  - ASNI, HSNI, AMNI, HMNI, PMNI, and PMU registers.
  - IDM, APU, and PAM subfeature registers.
  - FMU and FCU.

## Model limitations



Issues listed in this section have identifiers of the form [SDDKW-x]. These are for Arm internal references only.

- The following features are out of scope for the Fast Model, and will not be supported:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Power, clock and interrupt signals are not supported.
  - No support for AHB Locked transfers.
  - No support for IDM timeout detection.
  - No support for reorder buffers or Cyclic Dependency Avoidance Scheme (CDAS).
- APU limitations [SDDKW-82596]:
  - The base and size of address regions must be aligned to 4KiB even when `apuRegion4k` is false. A warning will be given and the region may not function properly due to PVBUS limitations.
  - `APU_ENABLE_RESET_STRAP` is not supported and `APU_CTRL.apu_enable` bit should be used to enable APU instead.
  - APU Region Locking is currently not supported. [SDDKW-80598]
  - `APUID` is not read from the transaction attributes, instead the model parameter `apu_subsystem_id` is used to specify the `subsystemID/APUID` of the requestor sending in a transaction through an `xSNI`. [SDDKW-81014]
  - If a transaction comes through an `xSNI` which does not have an APU and is routed to an `xMNI` which does have one, the APU in the `xMNI` treats the `APUID` as 0.
  - Since the `subsystemID/APUID` is not encoded in the transaction, components downstream to this model can't know the `APUID` of the requestor.
- Address map and bus traffic limitations [SDDKW-82595]:
  - Default target id cannot be configured through `sam_status` register. [SDDKW-77767] It can be provided through `default_tgtid_strap_i` model parameter.
  - Routers are not modeled. `xSNI` access only considers whether the target is defined in the `xSNI`'s memory map. [SDDKW-79760]

- No support for AMNI Address shuttering. [SDDKW-80419]
- Model supports 1, 2, and 4 targets for power-of-two stripe group hashing. No other striping functions/target combinations are supported.
- Hashing of stripe groups limited to a minimum granularity of 4096B. This is a DMI limitation of PVBus. See [Bus traffic in Fast Models](#) for information.
- No support for exclusive monitoring (ASNI/HSNI/AMNI/HMNI). [SDDKW-79385]
- xSNI access to CFGNI is not limited by router connectivity defined in the `mesh_config_file`. It considers only whether the xSNI has the CFGNI target defined in its memory map.
- Stripe limitations
  - ASNI striping to an HMNI target has not been tested.
  - Limited testing of stripe groups with different numbers of targets and granularities.
  - No Support for single target stripe.
  - There is no support for additional granularities, or the additional stripe group remap functionality.
  - Limited 8-way striping testing for static address maps. Not supported for programmable address maps.
- Remapping features not supported:
  - Tested 5 out of the maximum of 8 remap states.
  - The priority for multiple address remapping states.
  - One target remapped to a different target.
  - A single target remapped to a stripe group.
  - One stripe group remapped to a different stripe group.
  - A stripe group remapped to a single target.
- Programmer's view limitations [SDDKW-82591]:
  - 64-bit register accesses are not supported. [SDDKW-75304]
  - An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
  - No support for secure/root override for register accesses. [SDDKW-77158]
  - IDM registers are not tested. [SDDKW-77474]
  - No support for `idm_sreset_strap_i` functionality. [SDDKW-80451]
  - AMNI nodes in the model are interface-indifferent and the registers may not reflect the protocol version.
  - Error injection and error generation (RAS) are not supported. All error registers are **RAZ/WI**. [SDDKW-73411]
  - No support for MTE.
- Topology size limitations [SDDKW-82589]:
  - Maximum number of Voltage Domains is 32. The maximum tested is 1.

- Maximum number of Power Domains is 32. The maximum tested is 2.
- Maximum number of Clock Domains is 32. The maximum tested is 16.
- Maximum number of PMU nodes is 32 (1 per Clock Domain). The maximum tested is 15.
- Maximum number of Subordinate Network Interfaces (SNIs) is 128 (ASNI + HSNI). The maximum tested is 128 ASNIs and 9 HSNI.
- Maximum number of Manager Network Interfaces (MNIs) is 127 (AMNI + HMNI + PMNI). The maximum tested is 127 AMNIs, 127 PMNIs, and 9 HMNIs.
- FMU and FCU limitations:
  - Registers are readable and writeable. No other functionality is modeled.
- CMO limitations:
  - No software control for the CMO terminate response in register `AMNI_CONFIG_CTL`.
  - CMO Response control is not supported.
- A warm reset using static address maps has not been tested. [SDDKW-84005]
- RME limitations:
  - RME disable per interface is not supported. The disable only works when the topology parameter `axi/rmeSupport` for all interfaces is set to Disabled. Complete support to disable RME such as hiding root override regs has not been modeled or tested. [SDDKW-85413]
  - RME disable does not prevent REALM/ROOT transactions from flowing through interconnect. [SDDKW-81472]
- AMNI nodes in the model are interface-indifferent and registers do not reflect the protocol version.
- A reset after model startup does not reset the registers or address remap selections.
- No register visibility support for a debugger.
- r1 not supported:
  - AXI-Stream (TSNI and TMNI) not supported
  - AMNI and HMNI exclusive monitors not supported

### Iris and MTI instances for NOC\_S3

This model has the following Iris instances:

**Table 3-1138: NOC\_S3 Iris instances**

InstanceName	ComponentName
NOC_S3	NOC-S3
NOC_S3.decoder	PVBusMapper
NOC_S3.dsu_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-1139: NOC\_S3 MTI instances**

InstanceName	ComponentName
NOC_S3	NOC-S3
NOC_S3.decoder	PVBusMapper
NOC_S3.dsu_mapper	PVBusMapper

### Ports for NOC\_S3

**Table 3-1140: Ports**

Name	Protocol	Type	Description
idm_reset_signal_amni[127]	Signal	Master	IDM reset signals to AMNIs.
idm_reset_signal_asni[128]	Signal	Master	IDM reset signals to ASNIs.
idm_reset_signal_hmni[127]	Signal	Master	IDM reset signals to HMNIs.
idm_reset_signal_hsni[128]	Signal	Master	IDM reset signals to HSNIs.
idm_reset_signal_pmni[127]	Signal	Master	IDM reset signals to PMNIs.
pvbus_m_amni[127]	PVBus	Master	AMNI downstream ports.
pvbus_m_hmni[127]	PVBus	Master	HMNI downstream ports.
pvbus_m_pmni[127]	PVBus	Master	PMNI downstream ports.
pvbus_s_asni[128]	PVBus	Slave	ASNI upstream ports.
pvbus_s_hsni[128]	PVBus	Slave	HSNI upstream ports.
reset_in	Signal	Slave	Reset signal.

### Parameters for NOC\_S3

#### apu\_subsystem\_id

##### Type

string

##### Default value

"0"

APUID/SubsystemID of the component connected to each xSNI.

Specify the subsystem id of the component connected to each <x>SNI by using a format like:

```
<x>SNI<m>=<subsystemID0>,<y>SNI<n>=<subsystemID1>
```

The subsystemID is assumed to be 0 for any component connected to an <x>SNI that does not appear in this list.

#### default\_tgt\_id\_strap\_i

##### Type

string

**Default value**

```
""
```

Default Target ID input.

Specify the target id for each <x>SNI by using a format like:

```
<x>SNI<m>=<target_id0>,<y>SNI<n>=<target_id1>
```

CFGNI (Configuration Network Interface) is used as the default target if a certain <x>SNI does not appear in the list.

**mesh\_config\_file****Type**

```
string
```

**Default value**

```
""
```

**Description**

Name of a file containing mesh placement of NOC\_S3 components.

**mpam\_attributes****Type**

```
string
```

**Default value**

```
""
```

User-defined transform to be applied to bus attributes like `MasterID`, `ExtendedID` OR `UserFlags`, for MPAM Attributes encoded into bus attributes.

For example:

```
ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS
```

An empty string disables MPAM support.

**periphbase****Type**

```
int
```

**Default value**

```
0xffffffffffffffff
```

**Description**

Value for PERIPHBASE.

**print\_config**

**Type**  
bool

**Default value**  
0x0

**Description**  
Enables printing the config register addresses.

**print\_parser\_log**

**Type**  
bool

**Default value**  
0x0

**Description**  
Enables printing the yaml config parser log messages.

**revision**

**Type**  
string

**Default value**  
"r0p0"

**Description**  
Component revision. Currently supports r0p0.

**show\_banner**

**Type**  
uint64\_t

**Default value**  
2

Show component banner:

**0**  
supress entire banner

**1**  
suppress config file

**2+**  
show full banner

### 3.10.73 OTPW

One Time Programmable Memory Wrapper. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1141: IP revisions support**

Revision	Quality level
0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Iris and MTI instances for OTPW

This model has the following Iris instances:

**Table 3-1142: OTPW Iris instances**

InstanceName	ComponentName
OTPW	OTPW

#### Ports for OTPW

**Table 3-1143: Ports**

Name	Protocol	Type	Description
apb_host_in	<a href="#">PVBUS</a>	Slave	APB3 Subordinate Interface - Access to OTP memory or OTP emulation memory
apb_otp_out	<a href="#">PVBUS</a>	Master	APB3 Manager Interface - Access to OTP memory
apb_register_in	<a href="#">PVBUS</a>	Slave	APB3 Subordinate Interface - Access to OTPW registers
axi_emulated_otp_out	<a href="#">PVBUS</a>	Master	AXI Manager Interface - Access to emulated OTP memory
n_poreset_in	<a href="#">Signal</a>	Slave	ICLU power-on reset in
otp_alarm_out	<a href="#">Signal</a>	Master	Alarm output destined for the Security Alarm Manager
otp_int_out	<a href="#">Signal</a>	Master	Interrupt output destined for the processor
otp_otp_is_ready_out	<a href="#">Signal</a>	Master	Status signal indicating that the OTPW is ready

#### Parameters for OTPW

##### OTP\_SIZE\_IN\_WORDS

##### Type

int

##### Default value

0x1000

##### Description

Indicates the size of the OTP memory in words.



**diagnostics**

**Type**

int

**Default value**

0x2

**Description**

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

**3.10.74 PL011\_Uart**

ARM PrimeCell UART(PL011). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1144: IP revisions support

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Using in\_file and in\_file\_escape\_sequence parameters**


The UART reads input from `in_file`. If `in_file` contains a line beginning:

```
## WaitForPrompt <something-up-to-end-of-line>
```

then the UART stops reading from `in_file` until the prompt has appeared.

For example, if `in_file` contains the following lines, the UART outputs `ls` only after the root prompt appears:

```
## WaitForPrompt root #  
ls
```



**Note**

Use the parameter `in_file_escape_sequence` to set a different escape sequence to `##`.

**Using the untimed\_fifos parameter**

When the `untimed_fifos` parameter is false, characters of serial data are clocked to or from the SerialData port at a rate controlled by the `clk_in_ref` clock rate and the `baud-rate-divider` configuration of the UART clock. Enabling `untimed_fifos` permits serial data to be sent or received as fast as it can be generated or consumed. The modem control signals are still generated correctly,

so the UART is not able to transmit data faster than the receiving end can handle. For example, TelnetTerminal uses the CTS signal to avoid overflowing its TCP/IP buffer. See [TelnetTerminal](#).

### Differences between the model and the RTL

This component does not implement the DMA functionality of the PL011 PrimeCell.

### Iris and MTI instances for PL011\_Uart

This model has the following Iris instances:

**Table 3-1145: PL011\_Uart Iris instances**

InstanceName	ComponentName
PL011_Uart	PL011_Uart
PL011_Uart.busslave	PVBusSlave
PL011_Uart.clk_divider	ClockDivider
PL011_Uart.timer	ClockTimerThread
PL011_Uart.timer.timer	ClockTimerThread64
PL011_Uart.timer.timer.thread	SchedulerThread
PL011_Uart.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-1146: PL011\_Uart MTI instances**

InstanceName	ComponentName
PL011_Uart	<a href="#">PL011_Uart</a>
PL011_Uart.busslave	<a href="#">PVBusSlave</a>
PL011_Uart.clk_divider	<a href="#">ClockDivider</a>

### Ports for PL011\_Uart

**Table 3-1147: Ports**

Name	Protocol	Type	Description
clk_in_ref	<a href="#">ClockSignal</a>	Slave	Clock input, typically 14.745MHz, which sets the master transmit/receive rate.
intr	<a href="#">Signal</a>	Master	Interrupt signal.
pvbuss	<a href="#">PVBus</a>	Slave	Slave port for register access.
serial_out	<a href="#">SerialData</a>	Master	Serial input/output and control signals. Used to communicate with a serial device, such as a terminal.

### Parameters for PL011\_Uart

#### baud\_rate

##### Type

int

**Default value**

0x9600

**Description**

Baud rate.

**`clk_divider.div`****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**`clk_divider.mul`****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**`clock_rate`****Type**

int

**Default value**

0xe10000

**Description**

Clock rate for PL011.

**`enable_dc4`****Type**

bool

**Default value**

0x1

**Description**

Enable DC4 commands (try echo -e "help\024" in a Linux shell in a serial console).

**`flow_ctrl_mask_en`****Type**

bool

**Default value**

0x0

**Description**

Enable hardware flow control workaround which forcefully disables CTSen and RTSen bits in UARTCR register.

**halt****Type**

bool

**Default value**

0x0

**Description**

Halt instead of shutdown for shutdown\_on\_eot and shutdown\_tag.

**in\_file****Type**

string

**Default value**

""

**Description**

Input file for data to be read by the UART.

**in\_file\_escape\_sequence****Type**

string

**Default value**

"##"

**Description**

Input file escape sequence.

**out\_file****Type**

string

**Default value**

""

**Description**

Output file to hold data written by the UART (use '-' to send all output to stdout).

**revision****Type**

string

**Default value**

"r1p4"

**Description**

Revision to simulate.

**shutdown\_on\_eot****Type**

bool

**Default value**

0x0

**Description**

Shutdown simulation when a EOT (ASCII 4) char is transmitted (useful for regression tests when semihosting is not available).

**shutdown\_tag****Type**

string

**Default value**

""

**Description**

Shutdown simulation when a string is transmitted.

**toggle\_mti****Type**

string

**Default value**

""

**Description**

Start/stop token for any ToggleMTI source. Argument uses the JSON format: [{ "start": "START-TOKEN", "stop": "STOP-TOKEN" }] where 'START-TOKEN/END-TOKEN' are the corresponding start/stop tokens for toggling the trace plugins. Note that '\n' will be ignored if at start or end of the token. For additional information, use 'help' as the value of this parameter.

**uart\_enable****Type**

bool

**Default value**

0x0

**Description**

Enable uart when the system starts up. (clock\_rate and baud\_rate are only valid when this option is enabled.).

**unbuffered\_output**

**Type**

bool

**Default value**

0x0

**Description**

Unbuffered output.

**untimed\_fifos**

**Type**

bool

**Default value**

0x1

**Description**

Ignore the clock rate and transmit/receive serial data immediately.

**3.10.75 PL022\_SSP**

ARM PrimeCell Synchronous Serial Port(PL022). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:


**Table 3-1148: IP revisions support**

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Differences between the model and the RTL**

Although the PL022\_SSP component has clock input, it is not internally clock-driven. This is different to the hardware.



Note

This component is a preliminary release. It is not a fully-supported peripheral.

## Iris and MTI instances for PL022\_SSP

This model has the following Iris instances:

**Table 3-1149: PL022\_SSP Iris instances**

InstanceName	ComponentName
PL022_SSP	PL022_SSP
PL022_SSP.busslave	PVBusSlave
PL022_SSP.prescaler	ClockDivider

This model has the following MTI trace components:

**Table 3-1150: PL022\_SSP MTI instances**

InstanceName	ComponentName
PL022_SSP.busslave	PVBusSlave
PL022_SSP.prescaler	ClockDivider

## Ports for PL022\_SSP

**Table 3-1151: Ports**

Name	Protocol	Type	Description
clk	ClockSignal	Slave	Main PrimeCell SSP clock input.
clkin	ClockSignal	Slave	PrimeCell SSP clock input.
clkout	ClockSignal	Master	Clock output.
intr	Signal	Master	Interrupt signaling.
pvbus	PVBus	Slave	Slave port for connection to PV bus master/decoder.
rorintr	Signal	Master	Receive overrun interrupt.
rtintr	Signal	Master	Receive timeout interrupt. We don't implement time out interrupt.
rx_dma_port	PL080_DMAC_DmaPortProtocol	Master	PrimeCell SSP receive DMA port.
rx_d	Value	Slave	PrimeCell SSP receive data.
rxintr	Signal	Master	Receive FIFO service request port.
tx_dma_port	PL080_DMAC_DmaPortProtocol	Master	PrimeCell SSP transmit DMA port.
tx_d	Value	Master	PrimeCell SSP transmit data.
txintr	Signal	Master	Transmit FIFO service request.

## Parameters for PL022\_SSP

### prescaler.div

#### Type

int

#### Default value

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**prescaler.mul**

**Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

### 3.10.76 PL030\_RTC

ARM PrimeCell Real Time Clock(PL030). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1152: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Iris and MTI instances for PL030\_RTC

This model has the following Iris instances:

**Table 3-1153: PL030\_RTC Iris instances**

InstanceName	ComponentName
PL030_RTC	PL030_RTC
PL030_RTC.busslave	PVBusSlave
PL030_RTC.timer	ClockTimerThread
PL030_RTC.timer.timer	ClockTimerThread64
PL030_RTC.timer.timer.thread	SchedulerThread
PL030_RTC.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-1154: PL030\_RTC MTI instances**

InstanceName	ComponentName
PL030_RTC.busslave	PVBusSlave



## Ports for PL030\_RTC

**Table 3-1155: Ports**

Name	Protocol	Type	Description
clock	<a href="#">ClockSignal</a>	Slave	Clock input, typically 1MHz, driving master count rate.
intr	<a href="#">Signal</a>	Master	Interrupt signaling.
pvbust	<a href="#">PVBus</a>	Slave	Slave port for connection to PV bus master/decoder.

## 3.10.77 PL031\_RTC

ARM PrimeCell Real Time Clock(PL031). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1156: IP revisions support**

Revision	Quality level
r1p3	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About PL031\_RTC

This component can provide a basic alarm function or long time base counter.

It has no impact on the performance of a PV system when idle or counting down. The component only executes code when the counter expires or during bus accesses.

### Iris and MTI instances for PL031\_RTC

This model has the following Iris instances:

**Table 3-1157: PL031\_RTC Iris instances**

InstanceName	ComponentName
PL031_RTC	PL031_RTC
PL031_RTC.busslave	PVBusSlave
PL031_RTC.timer	ClockTimerThread
PL031_RTC.timer.timer	ClockTimerThread64
PL031_RTC.timer.timer.thread	SchedulerThread
PL031_RTC.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-1158: PL031\_RTC MTI instances**

InstanceName	ComponentName
PL031_RTC.busslave	<a href="#">PVBusSlave</a>

## Ports for PL031\_RTC

**Table 3-1159: Ports**

Name	Protocol	Type	Description
clock	<a href="#">ClockSignal</a>	Slave	Clock input, typically 1MHz, driving master count rate.
intr	<a href="#">Signal</a>	Master	Interrupt signaling.
pvbust	<a href="#">PVBus</a>	Slave	Slave port for connection to PV bus master/decoder.

## Parameters for PL031\_RTC

### RTCDR\_reset\_value

#### Type

int

#### Default value

0x0

#### Description

Reset value for RTCDR.

### RTCDR\_use\_current\_time

#### Type

bool

#### Default value

0x1

#### Description

Use current Unix/POSIX time for reset value for RTCDR. If true RTCDR\_reset\_value is ignored.

## 3.10.78 PL041\_AACI

ARM PrimeCell Advanced Audio CODEC Interface(PL041). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1160: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About PL041\_AACI

The PL041\_AACI component is designed to connect to an audio output component such as [AudioOut\\_File](#) or [AudioOut\\_SDL](#).

The ability to play audio through this component depends on the AudioOut component in use and on the performance requirements of the software running on the simulated system. The rate of FIFO draining is controlled by the audio output to which the component is connected. This might not correspond to the rate that would be expected from the reference clock.

This component also contains a minimal register model of the LM4529 secondary codec as implemented on development boards supplied by Arm.



This component is not a complete implementation of the AACI because the following functionality is not implemented:

- Audio input
- DMA access to FIFOs, rather than Programmed I/O
- Programming of the secondary codec through FIFOs rather than slot registers

### Iris and MTI instances for PL041\_AACI

This model has the following Iris instances:

**Table 3-1161: PL041\_AACI Iris instances**

InstanceName	ComponentName
PL041_AACI	PL041_AACI
PL041_AACI.busslave	PVBusSlave
PL041_AACI.timer	ClockTimerThread
PL041_AACI.timer.timer	ClockTimerThread64
PL041_AACI.timer.timer.thread	SchedulerThread
PL041_AACI.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-1162: PL041\_AACI MTI instances**

InstanceName	ComponentName
PL041_AACI.busslave	PVBusSlave

### Ports for PL041\_AACI

**Table 3-1163: Ports**

Name	Protocol	Type	Description
audio	AudioControl	Master	Used to communicate with an audio out device.
clk_in_ref	ClockSignal	Slave	Reference clock input, typically 25MH.
dma_rx	PL080_DMAC_DmaPortProtocol	Master	DMA receive port.
dma_tx	PL080_DMAC_DmaPortProtocol	Master	DMA transmit port.
irq	Signal	Master	Single IRQ output port.
pvbus	PVBus	Slave	Slave port for connection to PV bus master/decoder.

## Parameters for PL041\_AACI

### enabled

#### Type

bool

#### Default value

0x1

#### Description

Host interface connection enabled.

## 3.10.79 PL050\_KMI

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1164: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About PL050\_KMI

This model communicates with models of PS/2-like devices, for example a PS2Keyboard or PS2Mouse.

### Iris and MTI instances for PL050\_KMI

This model has the following Iris instances:

**Table 3-1165: PL050\_KMI Iris instances**

InstanceName	ComponentName
PL050_KMI	PL050_KMI
PL050_KMI.busslave	PVBusSlave
PL050_KMI.clk_divider	ClockDivider

This model has the following MTI trace components:

**Table 3-1166: PL050\_KMI MTI instances**

InstanceName	ComponentName
PL050_KMI.busslave	<a href="#">PVBusSlave</a>
PL050_KMI.clk_divider	<a href="#">ClockDivider</a>

## Ports for PL050\_KMI

**Table 3-1167: Ports**

Name	Protocol	Type	Description
clock	<a href="#">ClockSignal</a>	Slave	Clock input, typically 1MHz, which sets the master transmit/receive rate.
intr	<a href="#">Signal</a>	Master	Master port signaling completion of transmit or receive.
ps2device	<a href="#">PS2Data</a>	Slave	Used to communicate with a PS/2-like device.
pvbuss	<a href="#">PVBus</a>	Slave	Slave port for connection to PV bus master/decoder.

## Parameters for PL050\_KMI

### `clk_divider.div`

#### Type

int

#### Default value

0x1

#### Description

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

### `clk_divider.mul`

#### Type

int

#### Default value

0x1

#### Description

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

## 3.10.80 PL061\_GPIO

ARM PrimeCell General Purpose Input/Output(PL061). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1168: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## About PL061\_GPIO

This component provides eight programmable inputs or outputs. Ports of different widths can be created by multiple instantiation. In addition, an interrupt interface is provided to configure any number of pins as interrupt sources.

## Iris and MTI instances for PL061\_GPIO

This model has the following Iris instances:

**Table 3-1169: PL061\_GPIO Iris instances**

InstanceName	ComponentName
PL061_GPIO	PL061_GPIO
PL061_GPIO.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-1170: PL061\_GPIO MTI instances**

InstanceName	ComponentName
PL061_GPIO.busslave	PVBusSlave

## Ports for PL061\_GPIO

**Table 3-1171: Ports**

Name	Protocol	Type	Description
GPIO_In	Value	Slave	Input lines. 32-bit data in, only [7:0] is used.
GPIO_Intr	Signal	Master	Interrupt signal indicating to an interrupt controller that an interrupt occurred in one or more of the GPIO_In lines.
GPIO_MIS	Value	Master	Indicates the masked interrupt status. 32-bit data out , only [7:0] is used. NOT necessary, as the GPIOMIS can be read from address 0x418.
GPIO_Out	Value	Master	Output lines. 32-bit data out, only [7:0] is used.
pvbuss	PVBus	Slave	Slave port for register access.

## Parameters for PL061\_GPIO

### init\_inputs

#### Type

int

#### Default value

0x0

#### Description

Default input values [7:0].

## 3.10.81 PL080\_DMAC

ARM PrimeCell DMA Controller(PL080/081). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1172: IP revisions support**

Revision	Quality level
r1p3	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About PL080\_DMAC

This component provides 8 configurable DMA channels and 16 DMA ports for handshaking with peripherals. You can configure each channel to operate in one of eight flow control modes either under DMA control or the control of the source or destination peripheral. Transfers can occur on either master channel and can optionally be endian-converted on both source and destination transfers.

This component might have a significant impact on system performance in certain flow control modes. Channels configured for small bursts, or using single bursts, and with peripheral DMA handshaking could add significant overhead. The peripheral has not been fully optimized to make use of the advanced features of the PVBUS model.

### Iris and MTI instances for PL080\_DMAC

This model has the following Iris instances:

**Table 3-1173: PL080\_DMAC Iris instances**

InstanceName	ComponentName
PL080_DMAC	PL080_DMAC
PL080_DMAC.busmaster0	PVBusMaster
PL080_DMAC.busmaster1	PVBusMaster
PL080_DMAC.busslave	PVBusSlave
PL080_DMAC.timer	ClockTimerThread
PL080_DMAC.timer.timer	ClockTimerThread64
PL080_DMAC.timer.timer.thread	SchedulerThread
PL080_DMAC.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-1174: PL080\_DMAC MTI instances**

InstanceName	ComponentName
PL080_DMAC.busmaster0	PVBusMaster
PL080_DMAC.busmaster1	PVBusMaster
PL080_DMAC.busslave	PVBusSlave

## Ports for PL080\_DMAC

Table 3-1175: Ports

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	Clock signal to control DMA transfer rate.
dma_port[16]	<a href="#">PL080_DMAC_DmaPortProtocol</a>	Slave	request/response ports for communicating with devices.
interr	<a href="#">Signal</a>	Master	DMA error interrupt signal.
intr	<a href="#">Signal</a>	Master	Combined DMA error and terminal count signal.
inttc	<a href="#">Signal</a>	Master	DMA terminal count signal.
pvbush0_m	<a href="#">PVBush</a>	Master	Master bus interface 0 for DMA transfers.
pvbush1_m	<a href="#">PVBush</a>	Master	Master bus interface 1 for DMA transfers.
pvbush_s	<a href="#">PVBush</a>	Slave	Slave port for register accesses.
reset_in	<a href="#">Signal</a>	Slave	System reset.

## Parameters for PL080\_DMAC

**activate\_delay****Type**

int

**Default value**

0x0

**Description**

request delay.

**fifo\_size****Type**

int

**Default value**

0x10

**Description**

Channel FIFO size in bytes.

**generate\_clear****Type**

bool

**Default value**

0x0

**Description**

Generate clear response.



**max\_transfer****Type**

int

**Default value**

0x100

**Description**

Largest atomic transfer.

### 3.10.82 PL110\_CLCD

ARM PrimeCell Color LCD Controller(PL110). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1176: IP revisions support**

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About PL110\_CLCD

This implementation provides a register model of the LCD controller.

You can connect the model through a framebuffer port to a visualization component, for example, so that LCD output can be viewed.

The implementation is optimized for situations where the majority of the framebuffer does not change. For instance, displaying full-screen video results in significantly reduced performance. Rendering pixel data into an appropriate form for the framebuffer port (rasterization) can also take a significant amount of simulation time. If the pixel data are coming from a PVBusSlave region that has been configured as memory-like, rasterization only occurs in regions where memory contents are modified.

#### Iris and MTI instances for PL110\_CLCD

This model has the following Iris instances:

**Table 3-1177: PL110\_CLCD Iris instances**

InstanceName	ComponentName
PL110_CLCD	PL110_CLCD
PL110_CLCD.pl11x_clcd	PL11x_CLCD
PL110_CLCD.pl11x_clcd.busmaster	PVBusMaster
PL110_CLCD.pl11x_clcd.busslave	PVBusSlave
PL110_CLCD.pl11x_clcd.timer	ClockTimerThread

InstanceName	ComponentName
PL110_CLCD.pl11x_clcd.timer.timer	ClockTimerThread64
PL110_CLCD.pl11x_clcd.timer.timer.thread	SchedulerThread
PL110_CLCD.pl11x_clcd.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-1178: PL110\_CLCD MTI instances**

InstanceName	ComponentName
PL110_CLCD.pl11x_clcd.busmaster	PVBusMaster
PL110_CLCD.pl11x_clcd.busslave	PVBusSlave

### Ports for PL110\_CLCD

**Table 3-1179: Ports**

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Master clock input, typically 24MHz, to drive pixel clock timing.
control	Value	Slave	Auxiliary control register 1.
display	LCD	Master	Connection to visualization component.
intr	Signal	Master	Interrupt signaling for flyback events.
pvbus	PVBus	Slave	Slave port for register access.
pvbus_m	PVBus	Master	DMA port for video data.

### Parameters for PL110\_CLCD

#### **disable\_snooping\_dma**

##### Type

bool

##### Default value

0x0

##### Description

Disable DMA snooping.

#### **pixel\_double\_limit**

##### Type

int

##### Default value

0x12c

##### Description

Minimum LCD pixel width before display will be zoomed.

### 3.10.83 PL111\_CLCD

ARM PrimeCell Color LCD Controller(PL111). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1180: IP revisions support**

Revision	Quality level
r0p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About PL111\_CLCD

This component implements the hardware cursor support of the PL111\_CLCD, which is the main difference with PL110\_CLCD.

#### Iris and MTI instances for PL111\_CLCD

This model has the following Iris instances:

**Table 3-1181: PL111\_CLCD Iris instances**

InstanceName	ComponentName
PL111_CLCD	PL111_CLCD
PL111_CLCD.pl11x_clcd	PL11x_CLCD
PL111_CLCD.pl11x_clcd.busmaster	PVBusMaster
PL111_CLCD.pl11x_clcd.busslave	PVBusSlave
PL111_CLCD.pl11x_clcd.timer	ClockTimerThread
PL111_CLCD.pl11x_clcd.timer.timer	ClockTimerThread64
PL111_CLCD.pl11x_clcd.timer.timer.thread	SchedulerThread
PL111_CLCD.pl11x_clcd.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-1182: PL111\_CLCD MTI instances**

InstanceName	ComponentName
PL111_CLCD.pl11x_clcd.busmaster	<a href="#">PVBusMaster</a>
PL111_CLCD.pl11x_clcd.busslave	<a href="#">PVBusSlave</a>

#### Ports for PL111\_CLCD

**Table 3-1183: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	Master clock input, typically 24MHz, to drive pixel clock timing.
control	<a href="#">Value</a>	Slave	Auxiliary control register 1.
display	<a href="#">LCD</a>	Master	Connection to visualization component.

Name	Protocol	Type	Description
intr	Signal	Master	Interrupt signaling for flyback events.
pvbus	PVBus	Slave	Slave port for register access.
pvbus_m	PVBus	Master	DMA port for video data.

## Parameters for PL111\_CLCD

### **disable\_snooping\_dma**

#### Type

bool

#### Default value

0x0

#### Description

Disable DMA snooping.

### **pixel\_double\_limit**

#### Type

int

#### Default value

0x12c

#### Description

Minimum LCD pixel width before display will be zoomed.

## 3.10.84 PL180\_MCI

ARM PrimeCell Multimedia Card Interface (PL180). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1184: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About PL180\_MCI

When paired with an MMC card model, the PL180\_MCI component provides emulation of a flexible, persistent storage mechanism. See [MMC](#). The PL180\_MCI component fully models the registers of the corresponding PrimeCell, but supports a subset of the functionality of the PL180:

- The controller supports block mode transfers, but does not currently support streaming data transfer.

- The controller can be attached to a single MMC device. The MMC bus mode and SDIO modes of the PL180 PrimeCell are not supported.
- Command and Data timeouts are not simulated.
- Payload CRC errors are not simulated.
- The DMA interface present in the PL180 PrimeCell is not modeled.
- Minimal timing is implemented within the model.



At compile time, you can enable command tracing within the PL180\_MCI component by modifying the `PL180_TRACE` macro in the `MMC.lisa` file. This sends command and event trace to standard output. You can use this output to help diagnose device driver and controller-to-card protocol issues.

## Iris and MTI instances for PL180\_MCI

This model has the following Iris instances:

**Table 3-1185: PL180\_MCI Iris instances**

InstanceName	ComponentName
PL180_MCI	PL180_MCI
PL180_MCI.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-1186: PL180\_MCI MTI instances**

InstanceName	ComponentName
PL180_MCI.busslave	PVBusSlave

## Ports for PL180\_MCI

**Table 3-1187: Ports**

Name	Protocol	Type	Description
MCIINTR[2]	Signal	Master	Interrupts.
mmc_m	MMC_Protocol	Master	The MultiMediaCard (MMC) master port.
pvbus	PVBus	Slave	Slave port for register access.

## Parameters for PL180\_MCI

### pl180\_fifo\_depth

Type

int

Default value

0x10

**Description**  
PL180 FIFO Depth.

3.10.85 PL192\_VIC

ARM PrimeCell Vectored Interrupt Controller(PL192). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1188: IP revisions support

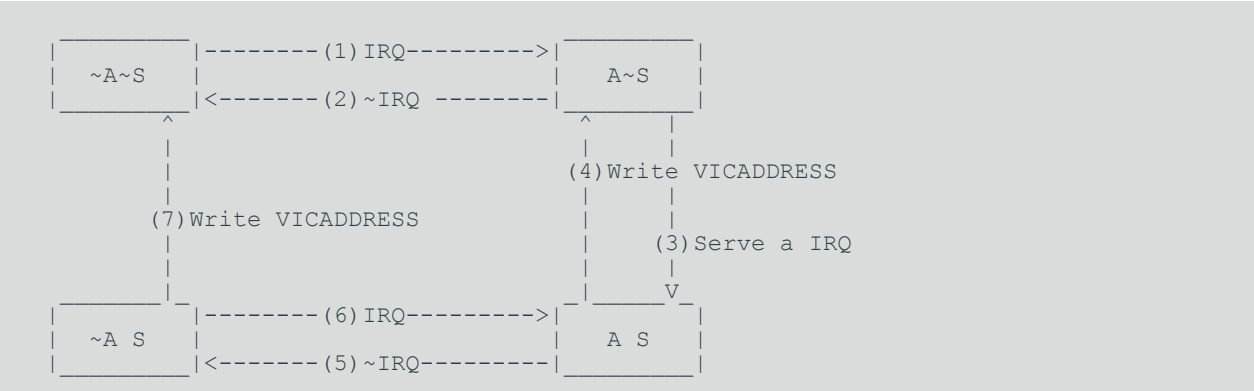
Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About PL192\_VIC

This component aggregates interrupts and generates interrupt signals for the Arm processor. When coupled with an Arm processor that provides a VIC port, routing to the appropriate interrupt handler can optionally be performed in hardware, reducing interrupt latency. The PL192\_VIC can also be daisy-chained with other PL192 VICs to permit more than 32 interrupts. The VIC supports hardware and software prioritization of interrupts.

This is the state transition diagram of a VIC 192 interrupt source:



- A  
The IRQ is active. It is in the irqServeList which is a sorted list of active IRQs that need to be served.
- $\sim A$   
Inactive IRQ. The corresponding input port is tied to low.
- S  
The IRQ is being served.
- $\sim S$   
The IRQ is waiting to be served

## Detailed Descriptions

1

An IRQ is asserted, and state changes from  $\sim A \sim S$  to  $A \sim S$ . The IRQ is inserted into a sorted list called `irqServeList` to wait for service. In this case, the corresponding input pin is tied to high.

2

The IRQ is deasserted. This can happen when the device does not want to keep IRQ active. For example, after continuously sending data, a UART can deassert IRQ to indicate stopping data transmission. In this case, even if the IRQ is in the stack and ready to be served it should be removed from the stack immediately.

3

When an IRQ that is at the top of the waiting stack is being served, the state changes from  $A \sim S$  to  $AS$ .

4

Writing to the `VICADDRESS` register indicates that the current served interrupt has been finished. However, the state of the IRQ could be still active. The device, such as a UART, that raised the IRQ could still want to generate a new IRQ to finish a task.

5

The IRQ is being served. Before finishing, the device deasserts the IRQ. The current IRQ will be removed from the top of the stack immediately, but it is still being served.

6

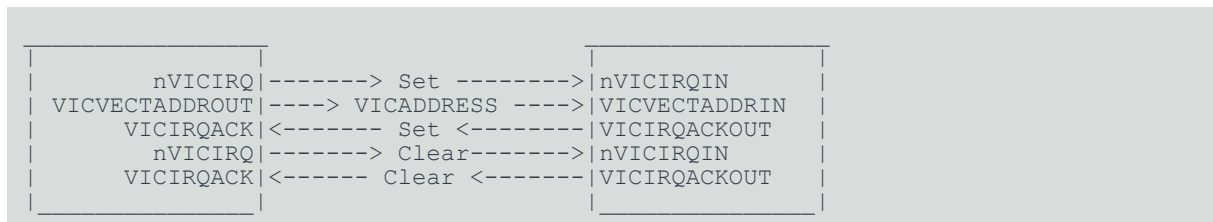
The IRQ is being served, and it is not in the stack. At this point, the device reasserts the IRQ and the state of IRQ changes from  $\sim AS$  to  $AS$ . In another words, the IRQ is reinserted into the stack.

7

An ISR writes to `VICADDRESS` to indicate the current IRQ has been served. Meanwhile the IRQ is deasserted by the device, the state of the VIC changes from  $\sim A S$  to  $\sim A \sim S$ .

The handshake when VIC is using VIC port to communicate. As this is an untimed model, it is not possible to model the timed nature of vector address passing accurately. There are two options offered:

1. Send the address just after the IRQ. This is closer to the hardware but requires that daisy-chained VICs repeatedly send their address as new, higher priority IRQs arrive, so may be slower:



2. Send the address during the ack. In this case, the ack ripples up through the VICs until it finds the IRQ and then the address ripples back down through the VICs, before the ack returns:



In both cases the ack clear is ignored by the VIC.

### Iris and MTI instances for PL192\_VIC

This model has the following Iris instances:

**Table 3-1189: PL192\_VIC Iris instances**

InstanceName	ComponentName
PL192_VIC	PL192_VIC
PL192_VIC.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-1190: PL192\_VIC MTI instances**

InstanceName	ComponentName
PL192_VIC.busslave	PVBusSlave

### Ports for PL192\_VIC

**Table 3-1191: Ports**

Name	Protocol	Type	Description
nVICFIQ	Signal	Master	Send out FIQ signal to the next level VIC or CPI.
nVICFIQIN	Signal	Slave	Used to receive FIQ signal when daisy chained.
nVICIRQ	Signal	Master	Send out IRQ signal to the next level VIC or procesessor.
nVICIRQIN	Signal	Slave	Used to receive IRQ signal when daisy chained.
pdbus	PVBus	Slave	Slave port for register access.
VICIntSource[32]	Signal	Slave	Interrupt source input sources.
VICIRQACK	Signal	Slave	Receive acknowledge signal from next level VIC or processor.
VICIRQACKOUT	Signal	Master	Used to send out acknowledge signals when daisy chained.
VICVECTADDRIN	ValueState	Slave	Used to receive vector address when daisy chained.
VICVECTADDRROUT	ValueState	Master	Used to send vector address to next level VIC or processor.



### 3.10.86 PL310\_L2CC

ARM PrimeCell Level 2 Cache Controller (PL310). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1192: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About PL310\_L2CC

The presence of additional on-chip secondary cache can improve performance when significant memory traffic is generated by the processor. A secondary cache assumes the existence of a Level 1, or primary, cache that is closely coupled or internal to the processor.

This component has two modes of operation, which are controlled by the `cache-state_modelled` parameter:

##### Register view

Cache control registers are present but the cache behavior is not modeled.

##### Functional model

Cache behavior is modeled.

Arm supports the use of the PL310 when connected to the Arm® Cortex®-A5 or Cortex-A9 processor.

This component implements the programmer-visible functionality of the PL310, and excludes some non-programmer visible features. The following features are implemented in the model:

- Physically addressed and physically tagged.
- Lockdown format C supported, for data and instructions. Lockdown format C is also known as way locking.
- Lockdown by line supported.
- Lockdown by master ID supported.
- Direct mapped to 16-way associativity, depending on the configuration and the use of lockdown registers. The associativity is configurable as 8 or 16.
- L2 cache available size can be 16 KB to 8 MB, depending on configuration and the use of the lockdown registers.
- Fixed line length of 32 bytes (8 words or 256 bits).
- Supports all of the AXI cache modes:
  - Write-through and write-back.
  - Read allocate, write allocate, read and write allocate.

- Force write-allocate option to always have cacheable writes allocated to L2 cache, for processors not supporting this mode.
- Normal memory non-cacheable shared reads are treated as cacheable non-allocatable. Normal memory non-cacheable shared writes are treated as cacheable write-through no write-allocate. There is an option, Shared Override, to override this behavior.
- TrustZone support, with the following features:
  - Non-Secure (NS) tag bit added in tag RAM and used for lookup in the same way as an address bit.
  - NS bit in Tag RAM used to determine security level of evictions to L3.
  - Restrictions for NS accesses for control, configuration, and maintenance registers to restrict access to secure data.
- Pseudo-Random victim selection policy. You can make this deterministic by using lockdown registers.
- Software option to enable exclusive cache configuration.
- Configuration registers accessible using address decoding in the component.
- Interrupt triggering in case of an error response when accessing L3.
- Maintenance operations.
- Prefetching capability.

The performance of this component depends on the configuration of the associated L1 caches and the mode it is in:

#### Register mode

No significant effect.

#### Functional mode with functional-mode L1

The addition of a functional L2 cache has minimal further impact on performance when running applications that are cache-bound.

#### Functional mode with a register-mode L1

There is a significant impact on system performance.



Note

Setting timing delays in this model does not impact the simulation speed. Generally, timing delays are only modeled for CPUs.

---

#### Differences between the model and the RTL

This model does not implement the following features, most of which are not relevant from a PV modeling point of view:

- There is no interface to the data and tag RAM as they are embedded in the model.
- Critical word first linefill is not supported, as it is not relevant for PV modeling.
- Buffers are not modeled.

- Outstanding accesses on slave and master ports cannot occur by design in a PV model as all transactions are atomic.
- Option to select one or two master ports and option to select one or two slave ports is not supported. Only one master port and one slave port are supported.
- Clock management and power modes are not supported, as they are not relevant for PV modeling.
- Wait, latency, clock enable, parity, and error support for data and tag RAMs are not included, as they are not relevant for PV modeling, and the data and tag RAMs embedded in the model cannot generate error responses.
- MBIST support is not included.
- Debug mode and debug registers are not supported.
- Test mode and scan chains are not supported.
- L2 cache event monitoring is not supported.
- Address filtering in the master ports is not supported.
- Performance counters are not supported.
- These Cortex-A9-related optimizations are not supported:
  - Prefetch hints
  - Full line of zero
  - Early write response
- Hazard detection is not required because of the atomic nature of the accesses in PV modeling and the fact that buffers are not modeled, therefore hazards cannot occur.
- Registers that belong to unimplemented features are accessible but do not have any functionality.

This model implements the following features differently to the hardware:

- Error handling. DECERR from the master port is mapped to SLVERR. Internal errors in cache RAM, for example parity errors, cannot happen in the model.
- Background cache operations do not occur in the background. They occur atomically.
- The LOCKDOWN\_BY\_LINE and LOCKDOWN\_BY\_MASTER parameter values are reflected in the CacheType register, but the feature is not switched off when the parameter is 0.
- This feature is additional:
  - Data RAM and Tag RAM are embedded in the model.

### Iris and MTI instances for PL310\_L2CC

This model has the following Iris instances:

**Table 3-1193: PL310\_L2CC Iris instances**

InstanceName	ComponentName
PL310_L2CC	PL310_L2CC

This model has the following MTI trace components:

**Table 3-1194: PL310\_L2CC MTI instances**

InstanceName	ComponentName
PL310_L2CC	PL310_L2CC

### Ports for PL310\_L2CC

**Table 3-1195: Ports**

Name	Protocol	Type	Description
DECERRINTR	Signal	Master	Decode error received on master port from L3.
ECNTRINTR	Signal	Master	Event Counter Overflow / Increment.
ERRRDINTR	Signal	Master	Error on L2 data RAM read.
ERRRTINTR	Signal	Master	Error on L2 tag RAM read.
ERRWDINTR	Signal	Master	Error on L2 data RAM write.
ERRWTINTR	Signal	Master	Error on L2 tag RAM write.
L2CCINTR	Signal	Master	Combined interrupt output.
PARRDINTR	Signal	Master	Parity error on L2 data RAM read.
PARRTINTR	Signal	Master	Parity error on L2 tag RAM read.
pvbus_m	PVBus	Master	Master port for connection to PV bus master/decoder.
pvbus_s	PVBus	Slave	Slave port for connection to PV bus master/decoder.
SLVERRINTR	Signal	Master	Slave error on master port from L3.

### Parameters for PL310\_L2CC

#### ASSOCIATIVITY

##### Type

int

##### Default value

0x0

##### Description

Associativity for Auxiliary Control Register.

#### CACHEID

##### Type

int

##### Default value

0x0

##### Description

Cache controller cache ID.

**CFGBIGEND****Type**

int

**Default value**

0x0

**Description**

Big-endian mode for accessing configuration registers out of reset.

**LOCKDOWN\_BY\_LINE****Type**

int

**Default value**

0x0

**Description**

Lockdown by line - value is reflected in CacheType register Bit 25, but the feature is not switched off when the parameter is 0.

**LOCKDOWN\_BY\_MASTER****Type**

int

**Default value**

0x0

**Description**

Lockdown by master - value is reflected in CacheType register Bit 26, but the feature is not switched off when the parameter is 0.

**REGFILEBASE****Type**

int

**Default value**

0x1f002000

**Description**

Base address for accessing configuration registers.

**WAYSIZE****Type**

int

**Default value**

0x1

**Description**

Size of ways for Auxiliary Control Register.

**cache-state\_modelled****Type**

bool

**Default value**

0x0

**Description**

Specifies whether real cache state is modelled (vs. register model).

**delay\_cache\_hit****Type**

int

**Default value**

0x0

**Description**

Cost to handle a cache hit.

**delay\_cache\_miss****Type**

int

**Default value**

0x0

**Description**

Cost to handle a cache miss.

**delay\_cache\_perbeat****Type**

int

**Default value**

0x0

**Description**

Cost to handle one beat of cache data movement.

### 3.10.87 PL330\_DMAC

ARM PrimeCell DMA Controller(PL330). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1196: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About PL330\_DMAC

The model uses a single LISA+ component but with a C++ model for each of the channels included in the LISA+ file. Enabled channels are kept on an enabled channels stack in priority order. When a channel state changes, re-arbitration takes place to make the highest (topmost) channel active.

Each transaction carries the identity of the requesting thread. This controller has up to eight channel threads and a manager thread. Each has an ID. In the hardware:

#### ID

AxID[3:0]

#### Identifying channels

0x0 - (numberOfChannels - 1)

#### Managers

numberOfChannels

For example, 0x0-0x7 and 0x8, respectively. The manager originates only instruction fetches, and the manager ID is also used for instruction fetches issued by the channels.

In the model, the identity of the requesting thread is encoded into each transaction using the low-order 16 bits of the Master ID field:

- Channel data: 0-7.
- Channel instruction fetch: 0xffff.
- Manager instruction fetch: 0xffff.

If a downstream component needs to know the IDs of bus masters that use either the low-order 16 bits or the label, use the label. The LabellerForDMA330 component shifts the low-order 16 bits into the label, while providing a degree of control over the label encoding. The example below maintains separate IDs for each data channel while using the correct hardware ID to identify instruction fetch for a DMA-330 with 8 channels:

```
p1330_dma : PL330_DMAC( "p_max_channels" = 8 );
dma_labeller : LabellerForDMA330(
    "dma330_discriminate_data_channels" = true,
    "dma330_s_instruction_label" = 8,
    "dma330_ns_instruction_label" = 8 );
p1330_dma.pvbus_m => dma_labeller.pvbus_s;
dma_labeller.pvbus_m => output_bus.pvbus_s;
```

### Iris and MTI instances for PL330\_DMAC

This model has the following Iris instances:

**Table 3-1197: PL330\_DMAC Iris instances**

InstanceName	ComponentName
PL330_DMAC	PL330_DMAC
PL330_DMAC.busmaster	PVBusMaster
PL330_DMAC.busslave	PVBusSlave
PL330_DMAC.busslave_ns	PVBusSlave
PL330_DMAC.timer	ClockTimerThread
PL330_DMAC.timer.timer	ClockTimerThread64
PL330_DMAC.timer.timer.thread	SchedulerThread
PL330_DMAC.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-1198: PL330\_DMAC MTI instances**

InstanceName	ComponentName
PL330_DMAC	PL330_DMAC
PL330_DMAC.busmaster	PVBusMaster
PL330_DMAC.busslave	PVBusSlave
PL330_DMAC.busslave_ns	PVBusSlave

## Ports for PL330\_DMAC

**Table 3-1199: Ports**

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock input.
irq_abort_master_port	Signal	Master	Undefined instruction or instruction error.
irq_master_port[32]	Signal	Master	Sets when DMASEV.
pvbuse_m	PVBus	Master	Master port for all memory accesses.
pvbuse_s	PVBus	Slave	Slave port for all register accesses (secure).
pvbuse_s_ns	PVBus	Slave	Slave port for all register accesses (non-secure).
reset_in	Signal	Slave	System reset.

## Parameters for PL330\_DMAC

### activate\_delay

#### Type

int

#### Default value

0x0

#### Description

request delay.



**fifo\_size****Type**

int

**Default value**

0x10

**Description**

Channel FIFO size in bytes.

**generate\_clear****Type**

bool

**Default value**

0x0

**Description**

Generate clear response.

**max\_transfer****Type**

int

**Default value**

0x100

**Description**

Largest atomic transfer.

**p\_axi\_bus\_width\_param****Type**

int

**Default value**

0x20

**Description**

AXI bus width.

**p\_buffer\_depth****Type**

int

**Default value**

0x10

**Description**

buffer depth.

**p\_cache\_line\_words****Type**

int

**Default value**

0x1

**Description**

number of words in a cache line.

**p\_cache\_lines****Type**

int

**Default value**

0x1

**Description**

number of cache lines.

**p\_controller\_boots****Type**

bool

**Default value**

0x1

**Description**

DMA boots from reset.

**p\_controller\_nsecure****Type**

bool

**Default value**

0x0

**Description**

Controller non-secure at reset (boot\_manager\_ns).

**p\_irq\_nsecure****Type**

int

**Default value**

0x0

**Description**

Interrupts non-secure at reset.

**p\_lsq\_read\_size****Type**

int

**Default value**

0x4

**Description**

LSQ read buffer depth.

**p\_lsq\_write\_size****Type**

int

**Default value**

0x4

**Description**

LSQ write buffer depth.

**p\_max\_channels****Type**

int

**Default value**

0x8

**Description**

virtual channels.

**p\_max\_irqs****Type**

int

**Default value**

0x20

**Description**

number of interrupts.

**p\_max\_periph****Type**

int

**Default value**

0x20

**Description**

number of peripheral interfaces.

**p\_perip\_request\_acceptance\_0****Type**

int

**Default value**

0x2

**Description**

Peripheral 0 request acceptance.

**p\_perip\_request\_acceptance\_1****Type**

int

**Default value**

0x2

**Description**

Peripheral 1 request acceptance.

**p\_perip\_request\_acceptance\_10****Type**

int

**Default value**

0x2

**Description**

Peripheral 10 request acceptance.

**p\_perip\_request\_acceptance\_11****Type**

int

**Default value**

0x2

**Description**

Peripheral 11 request acceptance.

**p\_perip\_request\_acceptance\_12****Type**

int

**Default value**

0x2

**Description**

Peripheral 12 request acceptance.

**p\_perip\_request\_acceptance\_13****Type**

int

**Default value**

0x2

**Description**

Peripheral 13 request acceptance.

**p\_perip\_request\_acceptance\_14****Type**

int

**Default value**

0x2

**Description**

Peripheral 14 request acceptance.

**p\_perip\_request\_acceptance\_15****Type**

int

**Default value**

0x2

**Description**

Peripheral 15 request acceptance.

**p\_perip\_request\_acceptance\_16****Type**

int

**Default value**

0x2

**Description**

Peripheral 16 request acceptance.

**p\_perip\_request\_acceptance\_17****Type**

int

**Default value**

0x2

**Description**

Peripheral 17 request acceptance.

**p\_perip\_request\_acceptance\_18****Type**

int

**Default value**

0x2

**Description**

Peripheral 18 request acceptance.

**p\_perip\_request\_acceptance\_19****Type**

int

**Default value**

0x2

**Description**

Peripheral 19 request acceptance.

**p\_perip\_request\_acceptance\_2****Type**

int

**Default value**

0x2

**Description**

Peripheral 2 request acceptance.

**p\_perip\_request\_acceptance\_20****Type**

int

**Default value**

0x2

**Description**

Peripheral 20 request acceptance.

**p\_perip\_request\_acceptance\_21****Type**

int

**Default value**

0x2

**Description**

Peripheral 21 request acceptance.

**p\_perip\_request\_acceptance\_22****Type**

int

**Default value**

0x2

**Description**

Peripheral 22 request acceptance.

**p\_perip\_request\_acceptance\_23****Type**

int

**Default value**

0x2

**Description**

Peripheral 23 request acceptance.

**p\_perip\_request\_acceptance\_24****Type**

int

**Default value**

0x2

**Description**

Peripheral 24 request acceptance.

**p\_perip\_request\_acceptance\_25****Type**

int

**Default value**

0x2

**Description**

Peripheral 25 request acceptance.

**p\_perip\_request\_acceptance\_26****Type**

int

**Default value**

0x2

**Description**

Peripheral 26 request acceptance.

**p\_perip\_request\_acceptance\_27****Type**

int

**Default value**

0x2

**Description**

Peripheral 27 request acceptance.

**p\_perip\_request\_acceptance\_28****Type**

int

**Default value**

0x2

**Description**

Peripheral 28 request acceptance.

**p\_perip\_request\_acceptance\_29****Type**

int

**Default value**

0x2

**Description**

Peripheral 29 request acceptance.

**p\_perip\_request\_acceptance\_3****Type**

int

**Default value**

0x2

**Description**

Peripheral 3 request acceptance.

**p\_perip\_request\_acceptance\_30****Type**

int

**Default value**

0x2

**Description**

Peripheral 30 request acceptance.



**p\_perip\_request\_acceptance\_31****Type**

int

**Default value**

0x2

**Description**

Peripheral 31 request acceptance.

**p\_perip\_request\_acceptance\_4****Type**

int

**Default value**

0x2

**Description**

Peripheral 4 request acceptance.

**p\_perip\_request\_acceptance\_5****Type**

int

**Default value**

0x2

**Description**

Peripheral 5 request acceptance.

**p\_perip\_request\_acceptance\_6****Type**

int

**Default value**

0x2

**Description**

Peripheral 6 request acceptance.

**p\_perip\_request\_acceptance\_7****Type**

int

**Default value**

0x2

**Description**

Peripheral 7 request acceptance.

**p\_perip\_request\_acceptance\_8****Type**

int

**Default value**

0x2

**Description**

Peripheral 8 request acceptance.

**p\_perip\_request\_acceptance\_9****Type**

int

**Default value**

0x2

**Description**

Peripheral 9 request acceptance.

**p\_periph\_nsecure****Type**

bool

**Default value**

0x0

**Description**

Peripherals non-secure at reset.

**p\_read\_issuing\_capability****Type**

int

**Default value**

0x1

**Description**

AXI read issuing capability.

**p\_reset\_pc****Type**

int

**Default value**

0x60000000

**Description**

DMA PC at reset.

**p\_write\_issuing\_capability**

**Type**  
int

**Default value**  
0x1

**Description**  
AXI write issuing capability.

**revision**

**Type**  
string

**Default value**  
"r0p0"

**Description**  
revision ID.

3.10.88 PL340\_DMC

ARM PrimeCell Dynamic Memory Controller(PL340). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1200: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About PL340\_DMC

This component provides an interface for up to four DRAM chips. The implementation also provides an APB interface to configure the controller behavior. You can access the registers through the APB interface.

Iris and MTI instances for PL340\_DMC

This model has the following Iris instances:

Table 3-1201: PL340\_DMC Iris instances

InstanceName	ComponentName
PL340_DMC	PL340_DMC
PL340_DMC.apb_slave	PVBusSlave
PL340_DMC.exclusive_monitor0	PVBusExclusiveMonitor
PL340_DMC.exclusive_monitor0.bus_mapper	PVBusMapper

InstanceName	ComponentName
PL340_DMC.exclusive_monitor1	PVBusExclusiveMonitor
PL340_DMC.exclusive_monitor1.bus_mapper	PVBusMapper
PL340_DMC.exclusive_monitor2	PVBusExclusiveMonitor
PL340_DMC.exclusive_monitor2.bus_mapper	PVBusMapper
PL340_DMC.exclusive_monitor3	PVBusExclusiveMonitor
PL340_DMC.exclusive_monitor3.bus_mapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-1202: PL340\_DMC MTI instances**

InstanceName	ComponentName
PL340_DMC.apb_slave	PVBusSlave
PL340_DMC.exclusive_monitor0	PVBusExclusiveMonitor
PL340_DMC.exclusive_monitor0.bus_mapper	PVBusMapper
PL340_DMC.exclusive_monitor1	PVBusExclusiveMonitor
PL340_DMC.exclusive_monitor1.bus_mapper	PVBusMapper
PL340_DMC.exclusive_monitor2	PVBusExclusiveMonitor
PL340_DMC.exclusive_monitor2.bus_mapper	PVBusMapper
PL340_DMC.exclusive_monitor3	PVBusExclusiveMonitor
PL340_DMC.exclusive_monitor3.bus_mapper	PVBusMapper

## Ports for PL340\_DMC

**Table 3-1203: Ports**

Name	Protocol	Type	Description
apb_interface	PVBus	Slave	Receive the apb config read/writes here.
axi_if_in[4]	PVBus	Slave	Receive the axi reads/writes here; up to four chips can be connected.
axi_if_out[4]	PVBus	Master	The output ports where the actual mem chips are connected.

## Parameters for PL340\_DMC

### IF\_CHIP0

#### Type

int

#### Default value

0xffffffffffffffff

#### Description

Set this parameter to 0 if memory is connected.

**IF\_CHIP1****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Set this parameter to 0 if memory is connected.

**IF\_CHIP2****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Set this parameter to 0 if memory is connected.

**IF\_CHIP3****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Set this parameter to 0 if memory is connected.

**MEMORY\_WIDTH****Type**

int

**Default value**

0x20

**Description**

Set this parameter to 0 if memory is connected.

**exclusive\_monitor0.apply\_access\_width\_criteria\_to\_non\_excl\_stores****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**exclusive\_monitor0.clear\_on\_strex\_address\_mismatch****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**exclusive\_monitor0.enable\_component****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**exclusive\_monitor0.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**exclusive\_monitor0.match\_secure\_state****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**exclusive\_monitor0.monitor\_access\_level****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

**`exclusive_monitor0.monitor_non_excl_stores`****Type**

bool

**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**`exclusive_monitor0.number_of_monitors`****Type**

int

**Default value**

0x8

**Description**

Number of monitors.

**`exclusive_monitor0.shareability_domain`****Type**

int

**Default value**

0x3

**Description**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

**`exclusive_monitor1.apply_access_width_criteria_to_non_excl_stores`****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**`exclusive_monitor1.clear_on_strex_address_mismatch`****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**`exclusive_monitor1.enable_component`****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**`exclusive_monitor1.exclusive_monitor_clear_on_atomic_from_same_master`****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**`exclusive_monitor1.match_secure_state`****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**`exclusive_monitor1.monitor_access_level`****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.



**exclusive\_monitor1.monitor\_non\_excl\_stores****Type**

bool

**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**exclusive\_monitor1.number\_of\_monitors****Type**

int

**Default value**

0x8

**Description**

Number of monitors.

**exclusive\_monitor1.shareability\_domain****Type**

int

**Default value**

0x3

**Description**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

**exclusive\_monitor2.apply\_access\_width\_criteria\_to\_non\_excl\_stores****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**exclusive\_monitor2.clear\_on\_strex\_address\_mismatch****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**`exclusive_monitor2.enable_component`****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**`exclusive_monitor2.exclusive_monitor_clear_on_atomic_from_same_master`****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**`exclusive_monitor2.match_secure_state`****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**`exclusive_monitor2.monitor_access_level`****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

**`exclusive_monitor2.monitor_non_excl_stores`****Type**

bool

**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**`exclusive_monitor2.number_of_monitors`****Type**

int

**Default value**

0x8

**Description**

Number of monitors.

**`exclusive_monitor2.shareability_domain`****Type**

int

**Default value**

0x3

**Description**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

**`exclusive_monitor3.apply_access_width_criteria_to_non_excl_stores`****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**`exclusive_monitor3.clear_on_strex_address_mismatch`****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**exclusive\_monitor3.enable\_component****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**exclusive\_monitor3.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**exclusive\_monitor3.match\_secure\_state****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**exclusive\_monitor3.monitor\_access\_level****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

**exclusive\_monitor3.monitor\_non\_excl\_stores****Type**

bool

**Default value**

0x0

**Description**  
Monitor non-exclusive stores from the same master.

**exclusive\_monitor3.number\_of\_monitors**

**Type**  
int

**Default value**  
0x8

**Description**  
Number of monitors.

**exclusive\_monitor3.shareability\_domain**

**Type**  
int

**Default value**  
0x3

**Description**  
Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

3.10.89 PL350\_SMC

ARM PrimeCell Static Memory Controller(PL350). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1204: IP revisions support

Revision	Quality level
r1p2	Full support
r2p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About PL350\_SMC

This component provides two memory interfaces. Each interface can be connected to a maximum of four memory devices, giving a total of eight inputs from the PVBUSDecoder and eight outputs to either SRAM or NAND devices. Only one kind of memory can be connected to a particular interface, either SRAM or NAND.

It provides a PVBUS slave to control the device behavior. A remap port is also provided to assist in remapping particular memory regions.

This component is optimized to have negligible impact on transaction performance, except when memory remap settings are changed, when there might be a significant effect.

### Iris and MTI instances for PL350\_SMC

This model has the following Iris instances:

**Table 3-1205: PL350\_SMC Iris instances**

InstanceName	ComponentName
PL350_SMC	PL350_SMC
PL350_SMC.addr_remapper	TZSwitch
PL350_SMC.addr_remapper.pvbus_mapper	PVBusMapper
PL350_SMC.apb_slave	PVBusSlave
PL350_SMC.exclusive_monitor0_0	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor0_0.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor0_1	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor0_1.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor0_2	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor0_2.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor0_3	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor0_3.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor1_0	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor1_0.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor1_1	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor1_1.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor1_2	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor1_2.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor1_3	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor1_3.bus_mapper	PVBusMapper
PL350_SMC.master_if0_0	PVBusMaster
PL350_SMC.master_if0_1	PVBusMaster
PL350_SMC.master_if0_2	PVBusMaster
PL350_SMC.master_if0_3	PVBusMaster
PL350_SMC.master_if1_0	PVBusMaster
PL350_SMC.master_if1_1	PVBusMaster
PL350_SMC.master_if1_2	PVBusMaster
PL350_SMC.master_if1_3	PVBusMaster
PL350_SMC.nand_remap_slave	PVBusSlave

This model has the following MTI trace components:

**Table 3-1206: PL350\_SMC MTI instances**

InstanceName	ComponentName
PL350_SMC.addr_remapper.pvbus_mapper	PVBusMapper
PL350_SMC.apb_slave	PVBusSlave
PL350_SMC.exclusive_monitor0_0	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor0_0.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor0_1	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor0_1.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor0_2	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor0_2.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor0_3	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor0_3.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor1_0	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor1_0.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor1_1	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor1_1.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor1_2	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor1_2.bus_mapper	PVBusMapper
PL350_SMC.exclusive_monitor1_3	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitor1_3.bus_mapper	PVBusMapper
PL350_SMC.master_if0_0	PVBusMaster
PL350_SMC.master_if0_1	PVBusMaster
PL350_SMC.master_if0_2	PVBusMaster
PL350_SMC.master_if0_3	PVBusMaster
PL350_SMC.master_if1_0	PVBusMaster
PL350_SMC.master_if1_1	PVBusMaster
PL350_SMC.master_if1_2	PVBusMaster
PL350_SMC.master_if1_3	PVBusMaster
PL350_SMC.nand_remap_slave	PVBusSlave

## Ports for PL350\_SMC

**Table 3-1207: Ports**

Name	Protocol	Type	Description
apb_interface	PVBus	Slave	This is where we expect to receive all the APB data which is used to read/write the device regs.
axi_chip_if0_in[4]	PVBus	Slave	This is where we expect to receive all the AXI data which is used to read/wrie NAND/RAM mem.
axi_chip_if0_out[4]	PVBus	Master	Master interface 0 to connect to SRAM/NAND.
axi_chip_if1_in[4]	PVBus	Slave	This is where we expect to receive all the AXI data which is used to read/wrie NAND/RAM mem.
axi_chip_if1_out[4]	PVBus	Master	Master interface 1 to connect to SRAM/NAND.

Name	Protocol	Type	Description
axi_remap	PVBus	Slave	This is the remap port that the designer needs to connect to zero.
irq_in_if0	Signal	Slave	Interrupt signals from devices connected on interface 0.
irq_in_if1	Signal	Slave	Interrupt signals from device connected on interface 1.
irq_out	Signal	Master	Interrupt port.
nand_remap_port	PVBus	Slave	Remaps the connected NAND port to 0x0.

## Parameters for PL350\_SMC

### IF0\_CHIP0

#### Type

bool

#### Default value

0x0

#### Description

Interface 0 chip 0 connected.

### IF0\_CHIP0\_BASE

#### Type

int

#### Default value

0x0

#### Description

Interface 0 chip 0 Base address.

### IF0\_CHIP0\_SIZE

#### Type

int

#### Default value

0x0

#### Description

Interface 0 chip 0 Size.

### IF0\_CHIP1

#### Type

bool

#### Default value

0x0

#### Description

Interface 0 chip 1 connected.



**IF0\_CHIP1\_BASE****Type**

int

**Default value**

0x0

**Description**

Interface 0 chip 1 Base address.

**IF0\_CHIP1\_SIZE****Type**

int

**Default value**

0x0

**Description**

Interface 0 chip 1 Size.

**IF0\_CHIP2****Type**

bool

**Default value**

0x0

**Description**

Interface 0 chip 2 connected.

**IF0\_CHIP2\_BASE****Type**

int

**Default value**

0x0

**Description**

Interface 0 chip 2 Base address.

**IF0\_CHIP2\_SIZE****Type**

int

**Default value**

0x0

**Description**

Interface 0 chip 2 Size.

**IF0\_CHIP3****Type**

bool

**Default value**

0x0

**Description**

Interface 0 chip 3 connected.

**IF0\_CHIP3\_BASE****Type**

int

**Default value**

0x0

**Description**

Interface 0 chip 3 Base address.

**IF0\_CHIP3\_SIZE****Type**

int

**Default value**

0x0

**Description**

Interface 0 chip 3 Size.

**IF0\_MEM\_TYPE\_PARAMETER****Type**

int

**Default value**

0x0

**Description**

Interface 0 Mem type.

**IF1\_CHIP0****Type**

bool

**Default value**

0x0

**Description**

Interface 1 chip 0 connected.

**IF1\_CHIP0\_BASE****Type**

int

**Default value**

0x0

**Description**

Interface 1 chip 0 Base address.

**IF1\_CHIP0\_SIZE****Type**

int

**Default value**

0x0

**Description**

Interface 1 chip 0 Size.

**IF1\_CHIP1****Type**

bool

**Default value**

0x0

**Description**

Interface 1 chip 1 connected.

**IF1\_CHIP1\_BASE****Type**

int

**Default value**

0x0

**Description**

Interface 1 chip 1 Base address.

**IF1\_CHIP1\_SIZE****Type**

int

**Default value**

0x0

**Description**

Interface 1 chip 1 Size.

**IF1\_CHIP2****Type**

bool

**Default value**

0x0

**Description**

Interface 1 chip 2 connected.

**IF1\_CHIP2\_BASE****Type**

int

**Default value**

0x0

**Description**

Interface 1 chip 2 Base address.

**IF1\_CHIP2\_SIZE****Type**

int

**Default value**

0x0

**Description**

Interface 1 chip 2 Size.

**IF1\_CHIP3****Type**

bool

**Default value**

0x0

**Description**

Interface 1 chip 3 connected.

**IF1\_CHIP3\_BASE****Type**

int

**Default value**

0x0

**Description**

Interface 1 chip 3 Base address.

**IF1\_CHIP3\_SIZE****Type**

int

**Default value**

0x0

**Description**

Interface 1 chip 3 Size.

**IF1\_MEM\_TYPE\_PARAMETER****Type**

int

**Default value**

0x0

**Description**

Interface 1 Mem type.

**PERIPH\_ID\_0****Type**

int

**Default value**

0x52

**Description**

Periph\_ID\_0 value.

**REMAP****Type**

int

**Default value**

0xffffffffffffffff

**Description**

Remap the device.

**addr\_remapper.normal****Type**

int

**Default value**

0x2

**Description**

Normal Port.

**addr\_remapper.secure****Type**

int

**Default value**

0x1

**Description**

Secure Port.

**exclusive\_monitor0\_0.apply\_access\_width\_criteria\_to\_non\_excl\_stores****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**exclusive\_monitor0\_0.clear\_on\_strex\_address\_mismatch****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**exclusive\_monitor0\_0.enable\_component****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**exclusive\_monitor0\_0.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**exclusive\_monitor0\_0.match\_secure\_state****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**exclusive\_monitor0\_0.monitor\_access\_level****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

**exclusive\_monitor0\_0.monitor\_non\_excl\_stores****Type**

bool

**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**exclusive\_monitor0\_0.number\_of\_monitors****Type**

int

**Default value**

0x8

**Description**

Number of monitors.

**exclusive\_monitor0\_0.shareability\_domain****Type**

int

**Default value**

0x3

**Description**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

**exclusive\_monitor0\_1.apply\_access\_width\_criteria\_to\_non\_excl\_stores****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**exclusive\_monitor0\_1.clear\_on\_strex\_address\_mismatch****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**exclusive\_monitor0\_1.enable\_component****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**exclusive\_monitor0\_1.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**exclusive\_monitor0\_1.match\_secure\_state****Type**

bool



**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**exclusive\_monitor0\_1.monitor\_access\_level****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

**exclusive\_monitor0\_1.monitor\_non\_excl\_stores****Type**

bool

**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**exclusive\_monitor0\_1.number\_of\_monitors****Type**

int

**Default value**

0x8

**Description**

Number of monitors.

**exclusive\_monitor0\_1.shareability\_domain****Type**

int

**Default value**

0x3

**Description**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

**exclusive\_monitor0\_2.apply\_access\_width\_criteria\_to\_non\_excl\_stores****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**exclusive\_monitor0\_2.clear\_on\_strex\_address\_mismatch****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**exclusive\_monitor0\_2.enable\_component****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**exclusive\_monitor0\_2.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**exclusive\_monitor0\_2.match\_secure\_state****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**exclusive\_monitor0\_2.monitor\_access\_level****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

**exclusive\_monitor0\_2.monitor\_non\_excl\_stores****Type**

bool

**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**exclusive\_monitor0\_2.number\_of\_monitors****Type**

int

**Default value**

0x8

**Description**

Number of monitors.

**exclusive\_monitor0\_2.shareability\_domain****Type**

int

**Default value**

0x3

**Description**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

**exclusive\_monitor0\_3.apply\_access\_width\_criteria\_to\_non\_excl\_stores****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**`exclusive_monitor0_3.clear_on_strex_address_mismatch`****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**`exclusive_monitor0_3.enable_component`****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**`exclusive_monitor0_3.exclusive_monitor_clear_on_atomic_from_same_master`****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**`exclusive_monitor0_3.match_secure_state`****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**`exclusive_monitor0_3.monitor_access_level`****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

**exclusive\_monitor0\_3.monitor\_non\_excl\_stores****Type**

bool

**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**exclusive\_monitor0\_3.number\_of\_monitors****Type**

int

**Default value**

0x8

**Description**

Number of monitors.

**exclusive\_monitor0\_3.shareability\_domain****Type**

int

**Default value**

0x3

**Description**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

**exclusive\_monitor1\_0.apply\_access\_width\_criteria\_to\_non\_excl\_stores****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**exclusive\_monitor1\_0.clear\_on\_strex\_address\_mismatch****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**`exclusive_monitor1_0.enable_component`****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**`exclusive_monitor1_0.exclusive_monitor_clear_on_atomic_from_same_master`****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**`exclusive_monitor1_0.match_secure_state`****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**`exclusive_monitor1_0.monitor_access_level`****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

**exclusive\_monitor1\_0.monitor\_non\_excl\_stores****Type**

bool

**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**exclusive\_monitor1\_0.number\_of\_monitors****Type**

int

**Default value**

0x8

**Description**

Number of monitors.

**exclusive\_monitor1\_0.shareability\_domain****Type**

int

**Default value**

0x3

**Description**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

**exclusive\_monitor1\_1.apply\_access\_width\_criteria\_to\_non\_excl\_stores****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**exclusive\_monitor1\_1.clear\_on\_strex\_address\_mismatch****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**exclusive\_monitor1\_1.enable\_component****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**exclusive\_monitor1\_1.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**exclusive\_monitor1\_1.match\_secure\_state****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**exclusive\_monitor1\_1.monitor\_access\_level****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

**exclusive\_monitor1\_1.monitor\_non\_excl\_stores****Type**

bool



**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**`exclusive_monitor1_1.number_of_monitors`****Type**

int

**Default value**

0x8

**Description**

Number of monitors.

**`exclusive_monitor1_1.shareability_domain`****Type**

int

**Default value**

0x3

**Description**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

**`exclusive_monitor1_2.apply_access_width_criteria_to_non_excl_stores`****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**`exclusive_monitor1_2.clear_on_strex_address_mismatch`****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**exclusive\_monitor1\_2.enable\_component****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**exclusive\_monitor1\_2.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**exclusive\_monitor1\_2.match\_secure\_state****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**exclusive\_monitor1\_2.monitor\_access\_level****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

**exclusive\_monitor1\_2.monitor\_non\_excl\_stores****Type**

bool

**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**`exclusive_monitor1_2.number_of_monitors`****Type**

int

**Default value**

0x8

**Description**

Number of monitors.

**`exclusive_monitor1_2.shareability_domain`****Type**

int

**Default value**

0x3

**Description**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

**`exclusive_monitor1_3.apply_access_width_criteria_to_non_excl_stores`****Type**

bool

**Default value**

0x1

**Description**

Apply the given exclusive store width matching criteria to non-exclusive stores.

**`exclusive_monitor1_3.clear_on_strex_address_mismatch`****Type**

bool

**Default value**

0x1

**Description**

Whether monitor is cleared when strex fails due to address mismatch.

**`exclusive_monitor1_3.enable_component`****Type**

bool

**Default value**

0x1

**Description**

Enable component.

**`exclusive_monitor1_3.exclusive_monitor_clear_on_atomic_from_same_master`****Type**

bool

**Default value**

0x1

**Description**

Monitor atomics from the same master.

**`exclusive_monitor1_3.match_secure_state`****Type**

bool

**Default value**

0x1

**Description**

Treat the secure state like an address bit.

**`exclusive_monitor1_3.monitor_access_level`****Type**

int

**Default value**

0x0

**Description**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

**`exclusive_monitor1_3.monitor_non_excl_stores`****Type**

bool

**Default value**

0x0

**Description**

Monitor non-exclusive stores from the same master.

**exclusive\_monitor1\_3.number\_of\_monitors**

**Type**  
int

**Default value**  
0x8

**Description**  
Number of monitors.

**exclusive\_monitor1\_3.shareability\_domain**

**Type**  
int

**Default value**  
0x3

**Description**  
Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

**revision**

**Type**  
string

**Default value**  
"r1p2"

**Description**  
Revision.

**3.10.90 PL350\_SMC\_NAND\_FLASH**

A NAND Flash implementation which works with PL350. This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1208: IP revisions support

Revision	Quality level
r1p2	Full support
r2p2	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**About PL350\_SMC\_NAND\_FLASH**

Program the component as you would the hardware.

## Iris and MTI instances for PL350\_SMC\_NAND\_FLASH

This model has the following Iris instances:

**Table 3-1209: PL350\_SMC\_NAND\_FLASH Iris instances**

InstanceName	ComponentName
PL350_SMC_NAND_FLASH	PL350_SMC_NAND_FLASH
PL350_SMC_NAND_FLASH.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-1210: PL350\_SMC\_NAND\_FLASH MTI instances**

InstanceName	ComponentName
PL350_SMC_NAND_FLASH.busslave	PVBusSlave

## Ports for PL350\_SMC\_NAND\_FLASH

**Table 3-1211: Ports**

Name	Protocol	Type	Description
irq	Signal	Master	Interrupt signaling.
pvbust	PVBus	Slave	Slave port for connection to PV bus master/decoder.

## Parameters for PL350\_SMC\_NAND\_FLASH

### DEVICE\_1

#### Type

int

#### Default value

0xec

#### Description

Device manufacturer code.

### DEVICE\_2

#### Type

int

#### Default value

0xda

#### Description

Device code.

### DEVICE\_3

#### Type

int

**Default value**

0x80

**Description**

Device 3rd cycle code.

**DEVICE\_4****Type**

int

**Default value**

0x15

**Description**

Device 4th cycle code.

**DEVICE\_NAME****Type**

string

**Default value**

"Samsung K9F1G08U0M"

**Description**

Device Name.

**NAND\_BLOCK\_COUNT****Type**

int

**Default value**

0x800

**Description**

number of blocks in the flash device.

**NAND\_FLASH\_SIZE****Type**

int

**Default value**

0x10800000

**Description**

flash size in byte.

**NAND\_PAGE\_COUNT\_PER\_BLOCK****Type**

int

**Default value**

0x40

**Description**

number of pages in each block.

**NAND\_PAGE\_SIZE**

**Type**

int

**Default value**

0x840

**Description**

page size.

**NAND\_SPARE\_SIZE\_PER\_PAGE**

**Type**

int

**Default value**

0x40

**Description**

Spare size per page.

**NAND\_VALID\_SIZE\_PER\_PAGE**

**Type**

int

**Default value**

0x800

**Description**

valid page size.

**3.10.91 PL370\_HDLCD**

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1212: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.



## Changes in 11.29.19

Ports added:

- `display_trace`



Too fast a pixel clock can slow the rest of the simulation.

## Iris and MTI instances for PL370\_HDLCD

This model has the following Iris instances:

**Table 3-1213: PL370\_HDLCD Iris instances**

InstanceName	ComponentName
PL370_HDLCD	PL370_HDLCD
PL370_HDLCD.busmaster	PVBusMaster
PL370_HDLCD.busslave	PVBusSlave
PL370_HDLCD.timer	ClockTimerThread
PL370_HDLCD.timer.timer	ClockTimerThread64
PL370_HDLCD.timer.timer.thread	SchedulerThread
PL370_HDLCD.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-1214: PL370\_HDLCD MTI instances**

InstanceName	ComponentName
PL370_HDLCD.busmaster	PVBusMaster
PL370_HDLCD.busslave	PVBusSlave

## Ports for PL370\_HDLCD

**Table 3-1215: Ports**

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Master clock input, typically 24MHz, to drive pixel clock timing.
display	LCD	Master	Connection to visualization component.
display_trace	FrameTracingProtocol	Master	Test/Debug Frame Capture connection.
intr	Signal	Master	Interrupt signaling line for flyback events.
pvbus	PVBus	Slave	Slave port for connection to PV bus master/decoder.
pvbus_m	PVBus	Master	DMA port for collecting video data from memory/framebuffer.

Parameters for PL370\_HDLCD

**diagnostics**

**Type**  
int

**Default value**  
0x0

**Description**  
Diagnostics level.

**disable\_snooping\_dma**

**Type**  
bool

**Default value**  
0x0

**Description**  
Disable DMA snooping.

**force\_frame\_rate**

**Type**  
int

**Default value**  
0x32

**Description**  
Force frame rate to the value of the parameter in frames per simulated second, regardless of the input clock. When 0, use the input clock as a pixel clock.

3.10.92 PL390\_GIC

Generic Interrupt Controller (PL390). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1216: IP revisions support

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

About PL390\_GIC

The GIC provides support for three interrupt types:

- Software Generated Interrupts (SGI)
- Private Peripheral Interrupts (PPI)
- Shared Peripheral Interrupts (SPI)

You can set:

- Security state for an interrupt
- Priority state for an interrupt
- Enabling or disabling state for an interrupt
- Processors that receive an interrupt

A processor interface consists of a pair of interfaces called `pvbuss_cpu` and `pvbuss_distributor`. The `enable_cx` and `match_cx` signals identify the originator of a transaction on `pvbuss_cpu`. Similarly, the `enable_dx` and `match_dx` signals identify the originator of a transaction on `pvbuss_distributor`. `X` corresponds to the number of a processor interface.



Note

To reduce compile time, the registers are not available by default. To activate them, uncomment either of the following statements in `PL390_GIC.lisa`:

```
// #define FEW_CADI_REGISTER
// #define ALL_CADI_REGISTER
```

## Iris and MTI instances for PL390\_GIC

This model has the following Iris instances:

**Table 3-1217: PL390\_GIC Iris instances**

InstanceName	ComponentName
PL390_GIC	PL390_GIC
PL390_GIC.busslave_cpu	PVBusSlave
PL390_GIC.busslave_distributor	PVBusSlave

This model has the following MTI trace components:

**Table 3-1218: PL390\_GIC MTI instances**

InstanceName	ComponentName
PL390_GIC.busslave_cpu	PVBusSlave
PL390_GIC.busslave_distributor	PVBusSlave

## Ports for PL390\_GIC

**Table 3-1219: Ports**

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Set preventing write accesses to security-critical configuration registers.

Name	Protocol	Type	Description
enable_c[8]	ValueState	Slave	Compared with masked PVBUS master id to select processor interface: (master_id & enable_c<n>) == match_c<n>.
enable_d[8]	ValueState	Slave	Compared with masked PVBUS master id to select distributor interface: (master_id & enable_d<n>) == match_d<n>.
legacy_nfiq[8]	Signal	Slave	Legacy FIQ interrupt for processor Interface <n>.
legacy_nirq[8]	Signal	Slave	Input interrupt signals.
match_c[8]	ValueState	Slave	Mask on the PVBUS master id to select processor interface: (master_id & enable_c<n>) == match_c<n>.
match_d[8]	ValueState	Slave	Mask on the PVBUS master id to select distributor interface: (master_id & enable_d<n>) == match_d<n>.
nfiq[8]	Signal	Master	Send out FIQ signal to processor <n>.
nirq[8]	Signal	Master	Send out IRQ signal to processor <n>.
ppi_c0[16]	Signal	Slave	Private peripheral interrupt for processor 0 (num_cpus> = 1).
ppi_c1[16]	Signal	Slave	Private peripheral interrupt for processor 1 (num_cpus> = 2).
ppi_c2[16]	Signal	Slave	Private peripheral interrupt for processor 2 (num_cpus> = 3).
ppi_c3[16]	Signal	Slave	Private peripheral interrupt for processor 3 (num_cpus> = 4).
ppi_c4[16]	Signal	Slave	Private peripheral interrupt for processor 4 (num_cpus> = 5).
ppi_c5[16]	Signal	Slave	Private peripheral interrupt for processor 5 (num_cpus> = 6).
ppi_c6[16]	Signal	Slave	Private peripheral interrupt for processor 6 (num_cpus> = 7).
ppi_c7[16]	Signal	Slave	Private peripheral interrupt for processor 7 (num_cpus> = 8).
pvbuss_cpu	PVBUS	Slave	Slave port for connection to processor interface.
pvbuss_distributor	PVBUS	Slave	Slave port for connection to distributor interface.
reset_in	Signal	Slave	Reset signal.
spi[988]	Signal	Slave	Shared peripheral interrupt inputs.

## Parameters for PL390\_GIC

### ARCHITECTURE\_VERSION

#### Type

int

#### Default value

0x1

#### Description

set architecture version in periph\_id register.

### AXI\_IF

#### Type

bool

#### Default value

0x1

**Description**

set interface type in peripheral identification register 8.

**C\_ID\_WIDTH****Type**

int

**Default value**

0x20

**Description**

width of the cpu interface master id.

**D\_ID\_WIDTH****Type**

int

**Default value**

0x20

**Description**

width of the distributor interface master id.

**ENABLE\_LEGACY\_FIQ****Type**

bool

**Default value**

0x1

**Description**

provide legacy fiq interrupt inputs.

**ENABLE\_LEGACY\_IRQ****Type**

bool

**Default value**

0x1

**Description**

provide legacy irq interrupt inputs.

**ENABLE\_PPI\_EDGE****Type**

bool

**Default value**

0x0

**Description**

ppi edge sensitive.

**ENABLE\_TRUSTZONE****Type**

bool

**Default value**

0x1

**Description**

support trustzone.

**INIT\_ENABLE\_C0****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_C0.

**INIT\_ENABLE\_C1****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_C1.

**INIT\_ENABLE\_C2****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_C2.

**INIT\_ENABLE\_C3****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_C3.

**INIT\_ENABLE\_C4****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_C4.

**INIT\_ENABLE\_C5****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_C5.

**INIT\_ENABLE\_C6****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_C6.

**INIT\_ENABLE\_C7****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_C7.

**INIT\_ENABLE\_D0****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_D0.

**INIT\_ENABLE\_D1****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_D1.

**INIT\_ENABLE\_D2****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_D2.

**INIT\_ENABLE\_D3****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_D3.

**INIT\_ENABLE\_D4****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_D4.

**INIT\_ENABLE\_D5****Type**

int

**Default value**

0xffffffff



**Description**

initial value of register ENABLE\_D5.

**INIT\_ENABLE\_D6****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_D6.

**INIT\_ENABLE\_D7****Type**

int

**Default value**

0xffffffff

**Description**

initial value of register ENABLE\_D7.

**INIT\_MATCH\_C0****Type**

int

**Default value**

0x0

**Description**

initial value of register MATCH\_C0.

**INIT\_MATCH\_C1****Type**

int

**Default value**

0x1

**Description**

initial value of register MATCH\_C1.

**INIT\_MATCH\_C2****Type**

int

**Default value**

0x2

**Description**

initial value of register MATCH\_C2.

**INIT\_MATCH\_C3****Type**

int

**Default value**

0x3

**Description**

initial value of register MATCH\_C3.

**INIT\_MATCH\_C4****Type**

int

**Default value**

0x4

**Description**

initial value of register MATCH\_C4.

**INIT\_MATCH\_C5****Type**

int

**Default value**

0x5

**Description**

initial value of register MATCH\_C5.

**INIT\_MATCH\_C6****Type**

int

**Default value**

0x6

**Description**

initial value of register MATCH\_C6.

**INIT\_MATCH\_C7****Type**

int

**Default value**

0x7

**Description**

initial value of register MATCH\_C7.

**INIT\_MATCH\_D0****Type**

int

**Default value**

0x0

**Description**

initial value of register MATCH\_D0.

**INIT\_MATCH\_D1****Type**

int

**Default value**

0x1

**Description**

initial value of register MATCH\_D1.

**INIT\_MATCH\_D2****Type**

int

**Default value**

0x2

**Description**

initial value of register MATCH\_D2.

**INIT\_MATCH\_D3****Type**

int

**Default value**

0x3

**Description**

initial value of register MATCH\_D3.

**INIT\_MATCH\_D4****Type**

int

**Default value**

0x4

**Description**

initial value of register MATCH\_D4.

**INIT\_MATCH\_D5****Type**

int

**Default value**

0x5

**Description**

initial value of register MATCH\_D5.

**INIT\_MATCH\_D6****Type**

int

**Default value**

0x6

**Description**

initial value of register MATCH\_D6.

**INIT\_MATCH\_D7****Type**

int

**Default value**

0x7

**Description**

initial value of register MATCH\_D7.

**NUM\_CPU****Type**

int

**Default value**

0x8

**Description**

number of cpu interfaces.

**NUM\_LSPI****Type**

int

**Default value**

0x1f

**Description**

number of lockable shared peripheral interrupts.

**NUM\_PPI****Type**

int

**Default value**

0x10

**Description**

number of peripheral interrupts.

**NUM\_PRIORITY\_LEVELS****Type**

int

**Default value**

0x100

**Description**

number of priority levels.

**NUM\_SGI****Type**

int

**Default value**

0x10

**Description**

number of software generated interrupts.

**NUM\_SPI****Type**

int

**Default value**

0x3dc

**Description**

number of shared peripheral interrupts.

### 3.10.93 PMU

PMU (Performance Monitoring Unit). This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1220: IP revisions support**

Revision	Quality level
0.0	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Changes in 11.29.19

Parameters added:

- `pm_64bit_ext`

### Iris and MTI instances for PMU

This model has the following Iris instances:

**Table 3-1221: PMU Iris instances**

InstanceName	ComponentName
PMU	PMU

### Ports for PMU

**Table 3-1222: Ports**

Name	Protocol	Type	Description
<code>apb_bus_s</code>	<a href="#">PVBUS</a>	Slave	-
<code>clk_in</code>	<a href="#">ClockSignal</a>	Slave	-

### Parameters for PMU

#### **diagnostics**

##### Type

int

##### Default value

0x2

##### Description

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

#### **feat\_rme**

##### Type

bool

##### Default value

0x0

##### Description

RME Support.

**is\_amu****Type**

bool

**Default value**

0x0

**Description**

AMU or PMU.

**num\_monitors****Type**

int

**Default value**

0x1

**Description**

Number of PMU monitors.

**pm\_64bit\_ext****Type**

bool

**Default value**

0x0

**Description**

64bit programmer view extension.

**pm\_dual\_page\_ext****Type**

bool

**Default value**

0x0

**Description**

Dual page in APB address space.

**pm\_edgedetect\_ext****Type**

bool

**Default value**

0x0

**Description**

Edge detect.

**pm\_export\_ext****Type**

bool

**Default value**

0x0

**Description**

Event output - exported event.

**pm\_fzo\_ext****Type**

bool

**Default value**

0x0

**Description**

Freeze on overflow.

**pm\_mpam\_filter\_ext****Type**

bool

**Default value**

0x0

**Description**

MPAM filtering.

**pm\_oac\_ext****Type**

bool

**Default value**

0x0

**Description**

Observability and access control.

**pm\_sos\_filter\_ext****Type**

bool

**Default value**

0x1

**Description**

Secure operating state filtering.



**pm\_sshot\_ext****Type**

bool

**Default value**

0x1

**Description**

Snapshot.

**pm\_threshold\_ext****Type**

bool

**Default value**

0x0

**Description**

Threshold.

**pm\_tro\_ext****Type**

bool

**Default value**

0x0

**Description**

Trace interface.

**pmevfiltr2\_present****Type**

bool

**Default value**

0x0

**Description**

Event filtering registers 2 present.

**pmevfiltr\_present****Type**

bool

**Default value**

0x0

**Description**

Event filtering registers present.

**pmimpdef\_present**

**Type**

bool

**Default value**

0x0

**Description**

Implementation defined register present.

**pmoflow\_present**

**Type**

bool

**Default value**

0x0

**Description**

Overflow interrupt present.

3.10.94 PPUMTWakerequest

Power Policy Unit (PPU) v8.2 Multi-threaded Core Wakerequest Logic. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1223: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Ports for PPUMTWakerequest

Table 3-1224: Ports

Name	Protocol	Type	Description
cpu_pchannel_m	PChannel	Master	-
ppu_pchannel_s	PChannel	Slave	-
thread_wake_request[2]	Signal	Slave	-
wakerequest	Signal	Master	-

Parameters for PPUMTWakerequest

**mt\_mode**

Type

bool

Default value

0x0

Description

Multi-threaded mode.

**thread0\_op\_mode\_bit**

Type

int

Default value

0x10

Description

Thread0 Operation Mode bit of DEVPACTIVE.

**thread1\_op\_mode\_bit**

Type

int

Default value

0x11

Description

Thread1 Operation Mode bit of DEVPACTIVE.

3.10.95 PPUv0

Power Policy Unit (PPU) v0.8 architectural model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1225: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Iris and MTI instances for PPUv0

This model has the following Iris instances:

**Table 3-1226: PPUv0 Iris instances**

InstanceName	ComponentName
PPUv0	PPUv0
PPUv0.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-1227: PPUv0 MTI instances**

InstanceName	ComponentName
PPUv0	PPUv0
PPUv0.busslave	PVBusSlave

### Ports for PPUv0

**Table 3-1228: Ports**

Name	Protocol	Type	Description
irq	Signal	Master	-
powerdown	Signal	Master	-
ppuhwstat	Value	Master	-
pvbus_s	PVBus	Slave	-
smpen	Signal	Slave	-
standbywfi	Signal	Slave	-
wakerequest	Signal	Slave	-

### Parameters for PPUv0

#### **default\_power\_state\_on**

##### Type

bool

##### Default value

0x0

##### Description

Default power state ON.

#### **device\_channels**

##### Type

int

##### Default value

0x0

##### Description

Number of device channels (0: P-Channel, 1-8: Q-Channels).

**dynamic\_off****Type**

bool

**Default value**

0x0

**Description**

Dynamic Off.

**dynamic\_on****Type**

bool

**Default value**

0x0

**Description**

Dynamic On.

**dynamic\_warm\_reset****Type**

bool

**Default value**

0x0

**Description**

Dynamic Warm Reset.

**full\_ret****Type**

int

**Default value**

0x0

**Description**

Full Retention (0: not supported, 1: static, 2: dynamic).

**func\_ret****Type**

int

**Default value**

0x0

**Description**

Functional Retention (0: not supported, 1: static, 2: dynamic).

**logic\_ret****Type**

int

**Default value**

0x0

**Description**

Logic Retention (0: not supported, 1: static, 2: dynamic).

**mem\_off****Type**

int

**Default value**

0x0

**Description**

Memory Off (0: not supported, 1: static, 2: dynamic).

**mem\_ret****Type**

int

**Default value**

0x0

**Description**

Memory Retention (0: not supported, 1: static, 2: dynamic).

**revision****Type**

string

**Default value**

"rOp0"

**Description**

Revision.

**use\_active\_signal****Type**

bool

**Default value**

0x0

**Description**

Use device-active signal.

### 3.10.96 PPUv1

Power Policy Unit (PPU) v1.1 architectural model. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1229: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About PPUv1

Software can determine which features the PPU supports by reading the PPU Identification Register 0, `PPU_IDR0` and the PPU Identification Register 1, `PPU_IDR1`.

The following power policies are offered by the PPU model, in order of increasing priority:

- Off.
- Emulated Off.
- Memory Retention.
- Emulated Memory Retention.
- Logic Retention.
- Full Retention.
- Memory Off.
- Functional Retention.
- On.
- Warm Reset.
- Debug Recovery Reset.

For the power mode transition rules, see [Arm Power Policy Unit Architecture Specification](#).

There are 16 operating mode values. The meaning of these values is specific to the device that is connected to the PPU. The operating mode can only be configured to change during a power transition of ON to ON.

The PPU model supports static and dynamic transitions on the P-Channel interface. It does not yet support Q-Channel.

`DEVPACTIVE` and `DEVSTATE` have the following bit encodings:

#### DEVPACTIVE bits [10:0]

Each bit indicates a required power mode.

**DEVPSTATE bits [3:0]**

The integer formed by this bitfield indicates a power mode.

**DEVPACTIVE bits [23:16]**

Operating mode. The interpretation of these bits depends on the `DEVPACTIVE` use model (Ladder or Independent).

**DEVPSTATE bits [7:4]**

The integer formed by this bitfield indicates an operating mode.

Communication over the Low Power Interface (`PREQUEST` and `PACTIVE`) uses blocking calls and does not model any delays. See [PChannel protocol] for further details.

For the AMBA Low Power Interface Specification Arm Q-Channel and P-Channel Interfaces, see [AMBA Low Power Interface Specification](#).

For static transitions, software sets the policy as the required power mode. The PPU then sends a `REQUEST` with the required power state to the attached device. The device can `ACCEPT` or `DENY` it. For dynamic transitions, software sets the policy as a minimum power mode. Based on whether the device has sent a signal using `DEVPACTIVE`, the PPU sends a `PREQUEST` with the required power state to the attached device. The device can `ACCEPT` or `DENY` it.

The PPU model is automatically reset by the simulation engine when the model starts up. Reset can also occur through the `reset_in` port. The PPU model is reset only when the signal value is `signal::Set`. Use `signal::Set` instead of zero, its integer value, to prevent unexpected behavior.

The `ppuhwstat` port notifies the power state change inside the PPU and the definition of each bit is the same as `DEVPACTIVE[10:0]`.

The `smpen` and `standbywfi` ports are defined in PPUv0 and are not supported in PPUv1.

**Differences between the model and the RTL**

- Q-Channel is not supported
- The PPU model has been validated with devices supporting only ON and OFF power modes. Arm has not tested the case where a connected device supports other power modes offered by the PPU.

**Iris and MTI instances for PPUv1**

This model has the following Iris instances:

**Table 3-1230: PPUv1 Iris instances**

InstanceName	ComponentName
PPUv1	PPUv1
PPUv1.busslave	PVBusSlave

This model has the following MTI trace components:



**Table 3-1231: PPUv1 MTI instances**

InstanceName	ComponentName
PPUv1	PPUv1
PPUv1.busslave	PVBusSlave

### Ports for PPUv1

**Table 3-1232: Ports**

Name	Protocol	Type	Description
dev_clk_en_out	Signal	Master	Domain clock enable
dev_emu_clk_en_out	Signal	Master	Domain emulated mode clock enable
dev_emu_isolaten_out	Signal	Master	Domain emulated isolation control.
dev_isolaten_out	Signal	Master	Domain isolation control.
dev_poresetn_out	Signal	Master	Domain power on reset
dev_ret_resetn_out	Signal	Master	Domain retention reset.
dev_warm_resetn_out	Signal	Master	Domain warm reset
devpactive	PChannel	Master	P-Channel port
irq	Signal	Master	PPU IRQ signal
powerdown	Signal	Master	Notify whether or not the PPU is in OFF state.
ppuhwstat	Value	Master	Notify the power state change inside the PPU. The definition of each bit is the same as DEVPACTIVE[10:0].
pvbuss_s	PVBus	Slave	PPU APB bus slave port
reset_in	Signal	Slave	PPU reset signal input
wakerequest	Signal	Slave	Input port for the wakerequest signal. It is ORed with PACTIVE[8] (ON) inside the PPU as input to PPU DEVPACTIVE[8] (ON). The "is_core_ppu" parameter controls whether there is additional logic to hold this signal until the PPU is in OFF/OFF_EMU state.

### Parameters for PPUv1

#### RevD\_support

##### Type

bool

##### Default value

0x1

##### Description

Whether to support Rev D locked IRQ.

#### bypass\_handshake

##### Type

bool

##### Default value

0x0

**Description**

Bypass pcsn handshake.

**dbg\_recov****Type**

int

**Default value**

0x0

**Description**

Debug Recovery Reset (0: not supported, 1: static mode only, 2: both dynamic & static mode).

**default\_op\_dyn\_en****Type**

bool

**Default value**

0x0

**Description**

Whether to enable operating mode dynamic transition by default.

**default\_op\_policy****Type**

int

**Default value**

0x0

**Description**

Default operating policy.

**default\_power\_state\_on****Type**

bool

**Default value**

0x0

**Description**

Default power state ON.

**default\_pwr\_dyn\_en****Type**

bool

**Default value**

0x0

**Description**

Whether to enable dynamic power mode transition by default.

**device\_channels****Type**

int

**Default value**

0x0

**Description**

Number of device channels (0: P-Channel, 1-8: Q-Channels).

**dynamic\_off****Type**

bool

**Default value**

0x0

**Description**

Dynamic Off.

**dynamic\_on****Type**

bool

**Default value**

0x0

**Description**

Dynamic On.

**dynamic\_warm\_reset****Type**

bool

**Default value**

0x0

**Description**

Dynamic Warm Reset.

**full\_ret****Type**

int

**Default value**

0x0

**Description**

Full Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

**func\_ret****Type**

int

**Default value**

0x0

**Description**

Functional Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

**is\_core\_ppu****Type**

bool

**Default value**

0x0

**Description**

PPU is core\_ppu type which means wake\_request would wait till PPU is OFF/OFF\_EMU.

**lock\_support****Type**

bool

**Default value**

0x1

**Description**

Whether to support OFF lock feature.

**logic\_ret****Type**

int

**Default value**

0x0

**Description**

Logic Retention (0: not supported, 1: static, 2: dynamic).

**mem\_off****Type**

int

**Default value**

0x0

**Description**

Memory Off (0: not supported, 1: static mode only, 2: both dynamic &amp; static mode).

**mem\_ret****Type**

int

**Default value**

0x0

**Description**

Memory Retention (0: not supported, 1: static mode only, 2: both dynamic &amp; static mode).

**mem\_ret\_emu****Type**

int

**Default value**

0x0

**Description**

Emulated Memory Retention (0: not supported, 1: static mode only, 2: both dynamic &amp; static mode).

**num\_opmode\_cfg****Type**

int

**Default value**

0x0

**Description**

Number of operating modes.

**off\_emu****Type**

int

**Default value**

0x0

**Description**

Emulated Off (0: not supported, 1: static mode only, 2: both dynamic &amp; static mode).

**off\_mem\_ret\_trans\_cfg****Type**

bool

**Default value**

0x0

**Description**

OFF to MEM\_RET direct transition configuration (0: not allowed, 1: allowed).

**op\_active\_cfg****Type**

int

**Default value**

0x0

**Description**

Operating mode active configuration (0: Ladder use model, 1: Independent user model).

**revision****Type**

string

**Default value**

"r1p1"

**Description**

Revision.

**use\_active\_signal****Type**

bool

**Default value**

0x0

**Description**

Use device-active signal.

### 3.10.97 PPUv1\_Cluster\_Wakerequest\_Logic

PPUv1 wake request stall logic. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1233: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Ports for PPUv1\_Cluster\_Wakerequest\_Logic

**Table 3-1234: Ports**

Name	Protocol	Type	Description
cluster_wake_request	Signal	Master	-
core_wake_request_in[8]	Signal	Slave	-
core_wake_request_out[8]	Signal	Master	-
ppuhwstat	Value	Slave	-
reset_in	Signal	Slave	-

### Parameters for PPUv1\_Cluster\_Wakerequest\_Logic

#### **core\_ppu\_wakerequest\_stall\_condition\_after\_reset**

##### Type

bool

##### Default value

0x0

##### Description

Set Stall Condition of Core WakeRequest (from GIC) for Core PPU after reset.

#### **disable\_core\_ppu\_wakerequest\_input\_stall**

##### Type

bool

##### Default value

0x0

##### Description

Disable wakerequest input stall of Core PPU. This feature is enabled by default to mimic the P-Channel request stall when Cluster PPU is in OFF.

#### **enable\_cluster\_wakeup\_if\_cluster\_on\_funcret**

##### Type

bool

##### Default value

0x1

**Description**

enable cluster wakeup logic. If it's disabled, core\_wake\_request\_in[x] will be directly connected to core\_wake\_request\_out[x] and cluster\_wake\_request port is disabled.

**3.10.98 RSE\_CPU\_Private\_Region**

RSE CPU processor private region. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1235: IP revisions support**

Revision	Quality level
1.46	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Iris and MTI instances for RSE\_CPU\_Private\_Region**

This model has the following Iris instances:

**Table 3-1236: RSE\_CPU\_Private\_Region Iris instances**

InstanceName	ComponentName
RSE_CPU_Private_Region	RSE_CPU_Private_Region
RSE_CPU_Private_Region.apb_nonsecure	PVBusSlave
RSE_CPU_Private_Region.apb_secure	PVBusSlave

This model has the following MTI trace components:

**Table 3-1237: RSE\_CPU\_Private\_Region MTI instances**

InstanceName	ComponentName
RSE_CPU_Private_Region.apb_nonsecure	<a href="#">PVBusSlave</a>
RSE_CPU_Private_Region.apb_secure	<a href="#">PVBusSlave</a>

**Ports for RSE\_CPU\_Private\_Region****Table 3-1238: Ports**

Name	Protocol	Type	Description
apb_nonsecure	<a href="#">PVBus</a>	Slave	-
apb_secure	<a href="#">PVBus</a>	Slave	secure & non-secure Subordinate APB Interface
reset_in	<a href="#">Signal</a>	Slave	Reset in signal



Parameters for RSE\_CPU\_Private\_Region

**CPUID\_RESET\_VALUE**

**Type**  
int

**Default value**  
0x0

**Description**  
CPUID Registers Reset Value.

**diagnostics**

**Type**  
int

**Default value**  
0x2

**Description**  
Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.10.99 RSE\_SystemControl

RSE System Control Registers. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1239: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Iris and MTI instances for RSE\_SystemControl**

This model has the following Iris instances:

Table 3-1240: RSE\_SystemControl Iris instances

InstanceName	ComponentName
RSE_SystemControl	RSE_SystemControl
RSE_SystemControl.busmaster	PVBusMaster
RSE_SystemControl.busslave	PVBusSlave
RSE_SystemControl.scp_rom_busmapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-1241: RSE\_SystemControl MTI instances**

InstanceName	ComponentName
RSE_SystemControl.busmaster	PVBusMaster
RSE_SystemControl.busslave	PVBusSlave
RSE_SystemControl.scp_rom_busmapper	PVBusMapper

## Ports for RSE\_SystemControl

**Table 3-1242: Ports**

Name	Protocol	Type	Description
boot_addr_out	Value_64	Master	Address when boot_en is enabled
boot_en_out	Signal	Master	Enables channel 0 to load first command after reset from boot_addr
boot_memattr_out	Value	Master	Memory attribute setting for the boot_addr
boot_shareattr_out	Value	Master	Shareability attribute for the boot_attr
busmaster_control	PVTransactionMaster	Master	-
cpu0_lockup_reset_request	Signal	Slave	-
cpu0_warm_reset_request	Signal	Slave	-
cpu1_lockup_reset_request	Signal	Slave	-
cpu1_warm_reset_request	Signal	Slave	-
cpu2_lockup_reset_request	Signal	Slave	-
cpu2_warm_reset_request	Signal	Slave	-
cpu3_lockup_reset_request	Signal	Slave	-
cpu3_warm_reset_request	Signal	Slave	-
cpuwait_out[4]	Signal	Master	-
dbgen_in	Signal	Slave	-
dbgen_out	Signal	Master	-
host_level_reset_request	Signal	Slave	-
initstvtor[4]	Value	Master	-
lcm_dcu_force_disable_out	Value	Master	LCM DCU Force disable signal
lcm_reset_request	Signal	Slave	-
lcm_sp_reset	Signal	Master	-
niden_in	Signal	Slave	-
niden_out	Signal	Master	-
nonsecure_watchdog_reset_request	Signal	Slave	-
pdcn_pvbus_m	PVBus	Master	-
po_reset	Signal	Master	-
pvbus_s	PVBus	Slave	-
reset_in	Signal	Slave	-
RESETREQ_in	StateSignal	Slave	-
RSE_PSI_STATUS[32]	Signal	Master	-

Name	Protocol	Type	Description
sam_reset_request	Signal	Slave	-
scp_cpu_reset	Signal	Master	-
scp_rom_access_pvbus_m	PVBus	Master	-
scp_rom_access_pvbus_s	PVBus	Slave	-
secure_watchdog_reset_request	Signal	Slave	-
slow_clock_watchdog_reset_request	Signal	Slave	-
software_reset_request	Signal	Slave	-
spiden_in	Signal	Slave	-
spiden_out	Signal	Master	-
spniden_in	Signal	Slave	-
spniden_out	Signal	Master	-
subsystem_hardware_reset_request	Signal	Slave	-
warm_reset	Signal	Master	-

## Parameters for RSE\_SystemControl

### COLDRESET\_MODE

#### Type

uint32\_t

#### Default value

0

Note: The external agents like BMC or ICU or reset controller can be used to reset the RSE in real hardware but in FVP we don't support these external agents Hence can't support full functionality of COLDRESET\_MODE=1 So keeping COLDRESET\_MODE default value to 0

### CPU0RSTREQENRST

#### Type

bool

#### Default value

0x0

#### Description

CPU 0 Warm Reset Request Enable Default Value.

### CPU0WAITRST

#### Type

bool

#### Default value

false

Note: If CPU0WAITRST is changed to 0x1, then RSE has to make use of DMA boot-flow and DMA has to release the RSE CPU wait signal, this is not currently implemented in FVP. So keeping CPU0WAITRST default value to false(0)

**CPU1WAITRST****Type**

bool

**Default value**

0x1

**Description**

Whether to hold cpu1 in reset at boot.

**CPU2WAITRST****Type**

bool

**Default value**

0x1

**Description**

Whether to hold cpu2 in reset at boot.

**CPU3WAITRST****Type**

bool

**Default value**

0x1

**Description**

Whether to hold cpu3 in reset at boot.

**DMA\_BOOT\_EN\_REG\_RESET****Type**

int

**Default value**

0x1

**Description**

Default Reset value of DMA\_BOOT\_EN register.

**GRETRREG\_RESET****Type**

int

**Default value**

0x0

**Description**

GRETREG Reset value.

**LCM\_DCU\_FORCE\_DISABLE\_REG\_RESET****Type**

int

**Default value**

0x55555555

**Description**

Default Reset value of LCM\_DCU\_FORCE\_DISABLE register.

**NUMCPU****Type**

int

**Default value**

0x1

**Description**

Number of Cortex-M CPU cores in the subsystem.

**NUMDMACHANNEL****Type**

int

**Default value**

0x2

**Description**

Number of DMA Channels.

**NUMVMBANK****Type**

int

**Default value**

0x2

**Description**

Number of Volatile Memory Banks.

**RSE\_DMA\_BOOT\_ADDR****Type**

int

**Default value**

0x407c00

**Description**

[25:0] bits of this parameter are mapped to bits [27:2] of DMA boot\_addr signal and the DMA\_BOOT\_ADDR register.

**RSE\_DMA\_BOOT\_REGION****Type**

int

**Default value**

0x1

**Description**

[3:0] bits of this parameter are mapped to bits [31:28] of DMA boot\_addr signal.

**SWRESETREQ\_BIT****Type**

int

**Default value**

0x5

**Description**

Software Reset Request Bit.

**allow\_lockup\_mask****Type**

bool

**Default value**

0x1

**Description**

Whether to allow masking of CPU lockup reset.

**diagnostics****Type**

int

**Default value**

0x0

**Description**

Diagnostics.

**reset\_vector\_addr**

**Type**  
int

**Default value**  
0x11000000

**Description**  
Reset Vector Address.

3.10.100 SMMUv3AEM

This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1243: IP revisions support

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Changes in 11.29.19

- Parameters added:
- hide\_warning\_ACCESSEN\_GPCEN\_set\_to\_1\_in\_a\_single\_write

About SMMUv3AEM

The SMMUv3 Architecture Envelope Model component is an architectural model that implements the SMMUv3.0 to SMMUv3.4 architectures for I/O virtualization of devices, except for the limitations listed below.

The SMMUv3 specifies that input addresses are conceptually 64 bits. The SMMUv3AEM model assumes that the input address is 64 bits. If the SoC has less than 64 bits as an input address bus then if the SoC wants to use the high address space (and use TTB1) then it must sign extend the address from the upstream peripherals to get to 64 bits.

Limitations

- No power control
- AMBA stash operations, destructive read and destructive hint operations are not supported on PVBUS and also are not supported by the device.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMC0_CEID0` fields. The PMU is intended for demonstration purposes only and for driver development.



Between 11.17 and 11.18, the point of triggering for events changed for 'TLB miss' and this might lead to an (architecturally valid) change in the values captured in some circumstances.

- It has limited RAS support configured by the `ras` parameter
- PCIe No\_snoop transactions are not supported
- SMMUV3.3 PMCG filtering by MPAM PARTID and PMG is not supported.
- SMMUV3.4 D128 descriptors are not supported.

## Notes

- A bad configuration renders the model inactive.
- Some configurations can be adjusted by configuration pins. These are only sampled at the negative edge of the reset pin. If you want to use these pins then you must drive them before sending a negative edge on the reset pin. During simulation reset the component driving them must also drive this transition again.
- Debug reads to the registers do not disturb the state.
- Writes to registers with Update flags, including debug writes, are ignored if the Update flag is already set to one.
- Debug and real accesses to the registers must be 32 bits or 64 bits.
- The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping used is remembered by the bus infrastructure and is used for subsequent sufficiently similar transactions without requiring intervention from the SMMU model, and is not traced, for instance.

The range of addresses that the mapping is valid for is determined by the SMMU model, depending on architectural and model-implementation details. However, as it is unaware of any sign-extension or the mapping that the SoC does, the SoC is responsible for subsequently narrowing the range of addresses for which this mapping is valid. Typically, this is done automatically when using PVBUSMapper.

- If the downstream system of the SMMU needs to distinguish the kinds of SMMU-originated accesses then:
  - For SMMU-originated writes, MSIs are issued using attributes determined by the parameter `msi_attribute_transform`, whilst Event queue writes are always issued with `ExtendedID=0, UserFlags=0, MasterID=0xFFFFffff` unless overridden by the parameter `tw_qs_attribute_transform`.
  - For SMMU-originated reads and compare-and-swap (CAS) operations, `tw_qs_attribute_transform` can be used.
  - If your system does table walks and queue accesses through TBUO (`separate_tw_msi_qs_port == false`), then care must be taken to distinguish table walk and queue traffic from normal translated traffic.



- If `SMMU_IDR1.TABLES_PRESET` or `SMMU_IDR1.QUEUES_PRESET` is set then see parameter `PRESET_REL_base_address` and the parameters it mentions.

Embedded implementations of the SMMU are allowed to have the queues/stream table in a 'close' RAM, either on-chip or in the SMMU itself. For the model, it is up to the integrator to supply this memory and for the SMMU model to be able to access it. Thus if the actual hardware has the memory built into the SMMU then it will be necessary for the integrator to wrap this model with a bus decoder and a memory model to more closely model the embedded implementation.

## Security State Determination (SSD)

The model uses the term SSD to mean the security state that the transaction, register, structure, or event belongs to. In the SMMUv3 architecture, the sideband signal `SEC_SID` holds this information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one-bit signal. With RME, in the hardware, it was expanded to two bits. To retain backwards compatibility in the model, `SEC_SID` remains as one-bit in the parameter `howto_identify`, but the second bit can be expressed with `SEC_SID_bit_1`, and its negative logic version `nSEC_SID_bit_1`.

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

## Parameters for parsing transaction attributes

Some of the parameters are strings that share a common parsing format for extracting from and placing information into the transaction's attributes.

They can extract and place information into the following fields of a transaction's attributes:

- `ManagerID64` (64 bits)
- `MasterID` (32 bits) (This is the lower 32 bits of `ManagerID64`)
- `ExtendedID` (64 bits)
- `UserFlags` (32 bits)

The string parameter is parsed as a comma-separated list of:

```
lhs_expression=rhs_expression
```

The `lhs_expression` and `rhs_expression` can be an entire symbol, for example:

```
StreamID
```

or a bit, or bit slice:

```
StreamID[16]
StreamID[31:20]
```

In `rhs_expression`, numeric constants (or slices of numeric constants) are also allowed:

```
StreamID[16]=1
StreamID[16]=10[2]
```

A single symbol might be assigned from multiple non-contiguous arrays of bits from a mix of different RHS symbols:

```
StreamID[16]=1, StreamID[31:30]=ExtendedID[1:0],
StreamID[15:0]=UserFlags[31:16], ...
```

In those cases where a left hand symbol can also appear on the right hand side, it is possible to swap bits and transform the symbol. For example, the following expression swaps bits 0 and 1 of the `ExtendedID`:

```
ExtendedID[0]=ExtendedID[1], ExtendedID[1]=ExtendedID[0]
```

Any bits of the attributes that have no transform specified are retained from the input.

The `lhs_expression` and `rhs_expression` must have the same bit width.

## Iris and MTI instances for SMMUv3AEM

This model has the following Iris instances:

**Table 3-1245: SMMUv3AEM Iris instances**

InstanceName	ComponentName
SMMUv3AEM	SMMUv3AEM
SMMUv3AEM.register_file[0]	PVBusSlave
SMMUv3AEM.service_request_tbu[0]	PVBusSlave
SMMUv3AEM.service_request_tbu[10]	PVBusSlave
SMMUv3AEM.service_request_tbu[11]	PVBusSlave
SMMUv3AEM.service_request_tbu[12]	PVBusSlave
SMMUv3AEM.service_request_tbu[13]	PVBusSlave
SMMUv3AEM.service_request_tbu[14]	PVBusSlave
SMMUv3AEM.service_request_tbu[15]	PVBusSlave
SMMUv3AEM.service_request_tbu[16]	PVBusSlave
SMMUv3AEM.service_request_tbu[17]	PVBusSlave
SMMUv3AEM.service_request_tbu[18]	PVBusSlave
SMMUv3AEM.service_request_tbu[19]	PVBusSlave

InstanceName	ComponentName
SMMUv3AEM.service_request_tbu[1]	PVBusSlave
SMMUv3AEM.service_request_tbu[20]	PVBusSlave
SMMUv3AEM.service_request_tbu[21]	PVBusSlave
SMMUv3AEM.service_request_tbu[22]	PVBusSlave
SMMUv3AEM.service_request_tbu[23]	PVBusSlave
SMMUv3AEM.service_request_tbu[24]	PVBusSlave
SMMUv3AEM.service_request_tbu[25]	PVBusSlave
SMMUv3AEM.service_request_tbu[26]	PVBusSlave
SMMUv3AEM.service_request_tbu[27]	PVBusSlave
SMMUv3AEM.service_request_tbu[28]	PVBusSlave
SMMUv3AEM.service_request_tbu[29]	PVBusSlave
SMMUv3AEM.service_request_tbu[2]	PVBusSlave
SMMUv3AEM.service_request_tbu[30]	PVBusSlave
SMMUv3AEM.service_request_tbu[31]	PVBusSlave
SMMUv3AEM.service_request_tbu[32]	PVBusSlave
SMMUv3AEM.service_request_tbu[33]	PVBusSlave
SMMUv3AEM.service_request_tbu[34]	PVBusSlave
SMMUv3AEM.service_request_tbu[35]	PVBusSlave
SMMUv3AEM.service_request_tbu[36]	PVBusSlave
SMMUv3AEM.service_request_tbu[37]	PVBusSlave
SMMUv3AEM.service_request_tbu[38]	PVBusSlave
SMMUv3AEM.service_request_tbu[39]	PVBusSlave
SMMUv3AEM.service_request_tbu[3]	PVBusSlave
SMMUv3AEM.service_request_tbu[40]	PVBusSlave
SMMUv3AEM.service_request_tbu[41]	PVBusSlave
SMMUv3AEM.service_request_tbu[42]	PVBusSlave
SMMUv3AEM.service_request_tbu[43]	PVBusSlave
SMMUv3AEM.service_request_tbu[44]	PVBusSlave
SMMUv3AEM.service_request_tbu[45]	PVBusSlave
SMMUv3AEM.service_request_tbu[46]	PVBusSlave
SMMUv3AEM.service_request_tbu[47]	PVBusSlave
SMMUv3AEM.service_request_tbu[48]	PVBusSlave
SMMUv3AEM.service_request_tbu[49]	PVBusSlave
SMMUv3AEM.service_request_tbu[4]	PVBusSlave
SMMUv3AEM.service_request_tbu[50]	PVBusSlave
SMMUv3AEM.service_request_tbu[51]	PVBusSlave
SMMUv3AEM.service_request_tbu[52]	PVBusSlave
SMMUv3AEM.service_request_tbu[53]	PVBusSlave
SMMUv3AEM.service_request_tbu[54]	PVBusSlave

InstanceName	ComponentName
SMMUv3AEM.service_request_tbu[55]	PVBusSlave
SMMUv3AEM.service_request_tbu[56]	PVBusSlave
SMMUv3AEM.service_request_tbu[57]	PVBusSlave
SMMUv3AEM.service_request_tbu[58]	PVBusSlave
SMMUv3AEM.service_request_tbu[59]	PVBusSlave
SMMUv3AEM.service_request_tbu[5]	PVBusSlave
SMMUv3AEM.service_request_tbu[60]	PVBusSlave
SMMUv3AEM.service_request_tbu[61]	PVBusSlave
SMMUv3AEM.service_request_tbu[62]	PVBusSlave
SMMUv3AEM.service_request_tbu[63]	PVBusSlave
SMMUv3AEM.service_request_tbu[6]	PVBusSlave
SMMUv3AEM.service_request_tbu[7]	PVBusSlave
SMMUv3AEM.service_request_tbu[8]	PVBusSlave
SMMUv3AEM.service_request_tbu[9]	PVBusSlave
SMMUv3AEM.tbu[0]	PVBusMapper
SMMUv3AEM.tbu[10]	PVBusMapper
SMMUv3AEM.tbu[11]	PVBusMapper
SMMUv3AEM.tbu[12]	PVBusMapper
SMMUv3AEM.tbu[13]	PVBusMapper
SMMUv3AEM.tbu[14]	PVBusMapper
SMMUv3AEM.tbu[15]	PVBusMapper
SMMUv3AEM.tbu[16]	PVBusMapper
SMMUv3AEM.tbu[17]	PVBusMapper
SMMUv3AEM.tbu[18]	PVBusMapper
SMMUv3AEM.tbu[19]	PVBusMapper
SMMUv3AEM.tbu[1]	PVBusMapper
SMMUv3AEM.tbu[20]	PVBusMapper
SMMUv3AEM.tbu[21]	PVBusMapper
SMMUv3AEM.tbu[22]	PVBusMapper
SMMUv3AEM.tbu[23]	PVBusMapper
SMMUv3AEM.tbu[24]	PVBusMapper
SMMUv3AEM.tbu[25]	PVBusMapper
SMMUv3AEM.tbu[26]	PVBusMapper
SMMUv3AEM.tbu[27]	PVBusMapper
SMMUv3AEM.tbu[28]	PVBusMapper
SMMUv3AEM.tbu[29]	PVBusMapper
SMMUv3AEM.tbu[2]	PVBusMapper
SMMUv3AEM.tbu[30]	PVBusMapper
SMMUv3AEM.tbu[31]	PVBusMapper

InstanceName	ComponentName
SMMUv3AEM.tbv[32]	PVBusMapper
SMMUv3AEM.tbv[33]	PVBusMapper
SMMUv3AEM.tbv[34]	PVBusMapper
SMMUv3AEM.tbv[35]	PVBusMapper
SMMUv3AEM.tbv[36]	PVBusMapper
SMMUv3AEM.tbv[37]	PVBusMapper
SMMUv3AEM.tbv[38]	PVBusMapper
SMMUv3AEM.tbv[39]	PVBusMapper
SMMUv3AEM.tbv[3]	PVBusMapper
SMMUv3AEM.tbv[40]	PVBusMapper
SMMUv3AEM.tbv[41]	PVBusMapper
SMMUv3AEM.tbv[42]	PVBusMapper
SMMUv3AEM.tbv[43]	PVBusMapper
SMMUv3AEM.tbv[44]	PVBusMapper
SMMUv3AEM.tbv[45]	PVBusMapper
SMMUv3AEM.tbv[46]	PVBusMapper
SMMUv3AEM.tbv[47]	PVBusMapper
SMMUv3AEM.tbv[48]	PVBusMapper
SMMUv3AEM.tbv[49]	PVBusMapper
SMMUv3AEM.tbv[4]	PVBusMapper
SMMUv3AEM.tbv[50]	PVBusMapper
SMMUv3AEM.tbv[51]	PVBusMapper
SMMUv3AEM.tbv[52]	PVBusMapper
SMMUv3AEM.tbv[53]	PVBusMapper
SMMUv3AEM.tbv[54]	PVBusMapper
SMMUv3AEM.tbv[55]	PVBusMapper
SMMUv3AEM.tbv[56]	PVBusMapper
SMMUv3AEM.tbv[57]	PVBusMapper
SMMUv3AEM.tbv[58]	PVBusMapper
SMMUv3AEM.tbv[59]	PVBusMapper
SMMUv3AEM.tbv[5]	PVBusMapper
SMMUv3AEM.tbv[60]	PVBusMapper
SMMUv3AEM.tbv[61]	PVBusMapper
SMMUv3AEM.tbv[62]	PVBusMapper
SMMUv3AEM.tbv[63]	PVBusMapper
SMMUv3AEM.tbv[6]	PVBusMapper
SMMUv3AEM.tbv[7]	PVBusMapper
SMMUv3AEM.tbv[8]	PVBusMapper
SMMUv3AEM.tbv[9]	PVBusMapper

This model has the following MTI trace components:

**Table 3-1246: SMMUv3AEM MTI instances**

InstanceName	ComponentName
SMMUv3AEM	SMMUv3AEM
SMMUv3AEM.register_file[0]	PVBusSlave
SMMUv3AEM.service_request_tbu[0]	PVBusSlave
SMMUv3AEM.service_request_tbu[10]	PVBusSlave
SMMUv3AEM.service_request_tbu[11]	PVBusSlave
SMMUv3AEM.service_request_tbu[12]	PVBusSlave
SMMUv3AEM.service_request_tbu[13]	PVBusSlave
SMMUv3AEM.service_request_tbu[14]	PVBusSlave
SMMUv3AEM.service_request_tbu[15]	PVBusSlave
SMMUv3AEM.service_request_tbu[16]	PVBusSlave
SMMUv3AEM.service_request_tbu[17]	PVBusSlave
SMMUv3AEM.service_request_tbu[18]	PVBusSlave
SMMUv3AEM.service_request_tbu[19]	PVBusSlave
SMMUv3AEM.service_request_tbu[1]	PVBusSlave
SMMUv3AEM.service_request_tbu[20]	PVBusSlave
SMMUv3AEM.service_request_tbu[21]	PVBusSlave
SMMUv3AEM.service_request_tbu[22]	PVBusSlave
SMMUv3AEM.service_request_tbu[23]	PVBusSlave
SMMUv3AEM.service_request_tbu[24]	PVBusSlave
SMMUv3AEM.service_request_tbu[25]	PVBusSlave
SMMUv3AEM.service_request_tbu[26]	PVBusSlave
SMMUv3AEM.service_request_tbu[27]	PVBusSlave
SMMUv3AEM.service_request_tbu[28]	PVBusSlave
SMMUv3AEM.service_request_tbu[29]	PVBusSlave
SMMUv3AEM.service_request_tbu[2]	PVBusSlave
SMMUv3AEM.service_request_tbu[30]	PVBusSlave
SMMUv3AEM.service_request_tbu[31]	PVBusSlave
SMMUv3AEM.service_request_tbu[32]	PVBusSlave
SMMUv3AEM.service_request_tbu[33]	PVBusSlave
SMMUv3AEM.service_request_tbu[34]	PVBusSlave
SMMUv3AEM.service_request_tbu[35]	PVBusSlave
SMMUv3AEM.service_request_tbu[36]	PVBusSlave
SMMUv3AEM.service_request_tbu[37]	PVBusSlave
SMMUv3AEM.service_request_tbu[38]	PVBusSlave
SMMUv3AEM.service_request_tbu[39]	PVBusSlave
SMMUv3AEM.service_request_tbu[3]	PVBusSlave

InstanceName	ComponentName
SMMUv3AEM.service_request_tbu[40]	PVBusSlave
SMMUv3AEM.service_request_tbu[41]	PVBusSlave
SMMUv3AEM.service_request_tbu[42]	PVBusSlave
SMMUv3AEM.service_request_tbu[43]	PVBusSlave
SMMUv3AEM.service_request_tbu[44]	PVBusSlave
SMMUv3AEM.service_request_tbu[45]	PVBusSlave
SMMUv3AEM.service_request_tbu[46]	PVBusSlave
SMMUv3AEM.service_request_tbu[47]	PVBusSlave
SMMUv3AEM.service_request_tbu[48]	PVBusSlave
SMMUv3AEM.service_request_tbu[49]	PVBusSlave
SMMUv3AEM.service_request_tbu[4]	PVBusSlave
SMMUv3AEM.service_request_tbu[50]	PVBusSlave
SMMUv3AEM.service_request_tbu[51]	PVBusSlave
SMMUv3AEM.service_request_tbu[52]	PVBusSlave
SMMUv3AEM.service_request_tbu[53]	PVBusSlave
SMMUv3AEM.service_request_tbu[54]	PVBusSlave
SMMUv3AEM.service_request_tbu[55]	PVBusSlave
SMMUv3AEM.service_request_tbu[56]	PVBusSlave
SMMUv3AEM.service_request_tbu[57]	PVBusSlave
SMMUv3AEM.service_request_tbu[58]	PVBusSlave
SMMUv3AEM.service_request_tbu[59]	PVBusSlave
SMMUv3AEM.service_request_tbu[5]	PVBusSlave
SMMUv3AEM.service_request_tbu[60]	PVBusSlave
SMMUv3AEM.service_request_tbu[61]	PVBusSlave
SMMUv3AEM.service_request_tbu[62]	PVBusSlave
SMMUv3AEM.service_request_tbu[63]	PVBusSlave
SMMUv3AEM.service_request_tbu[6]	PVBusSlave
SMMUv3AEM.service_request_tbu[7]	PVBusSlave
SMMUv3AEM.service_request_tbu[8]	PVBusSlave
SMMUv3AEM.service_request_tbu[9]	PVBusSlave
SMMUv3AEM.tb[0]	PVBusMapper
SMMUv3AEM.tb[10]	PVBusMapper
SMMUv3AEM.tb[11]	PVBusMapper
SMMUv3AEM.tb[12]	PVBusMapper
SMMUv3AEM.tb[13]	PVBusMapper
SMMUv3AEM.tb[14]	PVBusMapper
SMMUv3AEM.tb[15]	PVBusMapper
SMMUv3AEM.tb[16]	PVBusMapper
SMMUv3AEM.tb[17]	PVBusMapper

InstanceName	ComponentName
SMMUv3AEM.tbv[18]	PVBusMapper
SMMUv3AEM.tbv[19]	PVBusMapper
SMMUv3AEM.tbv[1]	PVBusMapper
SMMUv3AEM.tbv[20]	PVBusMapper
SMMUv3AEM.tbv[21]	PVBusMapper
SMMUv3AEM.tbv[22]	PVBusMapper
SMMUv3AEM.tbv[23]	PVBusMapper
SMMUv3AEM.tbv[24]	PVBusMapper
SMMUv3AEM.tbv[25]	PVBusMapper
SMMUv3AEM.tbv[26]	PVBusMapper
SMMUv3AEM.tbv[27]	PVBusMapper
SMMUv3AEM.tbv[28]	PVBusMapper
SMMUv3AEM.tbv[29]	PVBusMapper
SMMUv3AEM.tbv[2]	PVBusMapper
SMMUv3AEM.tbv[30]	PVBusMapper
SMMUv3AEM.tbv[31]	PVBusMapper
SMMUv3AEM.tbv[32]	PVBusMapper
SMMUv3AEM.tbv[33]	PVBusMapper
SMMUv3AEM.tbv[34]	PVBusMapper
SMMUv3AEM.tbv[35]	PVBusMapper
SMMUv3AEM.tbv[36]	PVBusMapper
SMMUv3AEM.tbv[37]	PVBusMapper
SMMUv3AEM.tbv[38]	PVBusMapper
SMMUv3AEM.tbv[39]	PVBusMapper
SMMUv3AEM.tbv[3]	PVBusMapper
SMMUv3AEM.tbv[40]	PVBusMapper
SMMUv3AEM.tbv[41]	PVBusMapper
SMMUv3AEM.tbv[42]	PVBusMapper
SMMUv3AEM.tbv[43]	PVBusMapper
SMMUv3AEM.tbv[44]	PVBusMapper
SMMUv3AEM.tbv[45]	PVBusMapper
SMMUv3AEM.tbv[46]	PVBusMapper
SMMUv3AEM.tbv[47]	PVBusMapper
SMMUv3AEM.tbv[48]	PVBusMapper
SMMUv3AEM.tbv[49]	PVBusMapper
SMMUv3AEM.tbv[4]	PVBusMapper
SMMUv3AEM.tbv[50]	PVBusMapper
SMMUv3AEM.tbv[51]	PVBusMapper
SMMUv3AEM.tbv[52]	PVBusMapper



InstanceName	ComponentName
SMMUv3AEM.tbv[53]	PVBusMapper
SMMUv3AEM.tbv[54]	PVBusMapper
SMMUv3AEM.tbv[55]	PVBusMapper
SMMUv3AEM.tbv[56]	PVBusMapper
SMMUv3AEM.tbv[57]	PVBusMapper
SMMUv3AEM.tbv[58]	PVBusMapper
SMMUv3AEM.tbv[59]	PVBusMapper
SMMUv3AEM.tbv[5]	PVBusMapper
SMMUv3AEM.tbv[60]	PVBusMapper
SMMUv3AEM.tbv[61]	PVBusMapper
SMMUv3AEM.tbv[62]	PVBusMapper
SMMUv3AEM.tbv[63]	PVBusMapper
SMMUv3AEM.tbv[6]	PVBusMapper
SMMUv3AEM.tbv[7]	PVBusMapper
SMMUv3AEM.tbv[8]	PVBusMapper
SMMUv3AEM.tbv[9]	PVBusMapper

## Ports for SMMUv3AEM

**Table 3-1247: Ports**

Name	Protocol	Type	Description
axi_stream_msi_addr_to_match_s	Value_64	Slave	Any SMMU-originated MSI which exactly matches the address in this port will be sent through the AXI stream port axi_stream_msi_m which is usually connected to the GIC through axi_stream_msi_s. As MSIs are 32 bit aligned then if bits[1:0] != 0 or the address is above OAS then this is effectively disabled. The parameter smmu_msi_device_id is the DeviceID to send on the interface. See also the parameters: axi_stream_msi_TID and axi_stream_msi_TDEST. The default value of this port is set by the parameter axi_stream_msi_addr_to_match. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
axi_stream_msi_m	PVBus	Master	Manager port used for sending SMMU originated MSIs directly to the GIC when axi_stream_msi_enabled == true
clk_in	ClockSignal	Slave	Clock signal
conf_reset_of_SMMU_GBPA_ABORT	Signal	Slave	System reset value of SMMU_GBPA.ABORT. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
conf_reset_of_SMMU_S_GBPA_ABORT	Signal	Slave	System reset value of SMMU_S_GBPA.ABORT. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
conf_system_supports_bgptm	Signal	Slave	System supports broadcast TLBI PAALL and TLBI RPA for supporting RME. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.

Name	Protocol	Type	Description
conf_system_supports_btm	Signal	Slave	System supports BTM and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If BTM (Broadcast Table Maintenance) is not supported then DVM messages will be ignored. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
conf_system_supports_cohacc	Signal	Slave	System supports COHACC and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If COHACC is set then page walks and SMMU-generated accesses will have the required shareability set, otherwise they will be marked as non-shareable. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
conf_system_supports_httu	Signal	Slave	System supports HTTU and will be reflected in the IDR registers. See parameter support_for_httu_when_starts_disallowed for the use of this signal. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
conf_system_supports_sev	Signal	Slave	System supports SEV and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
identify	SMMUv3AEMIdentifyProtocol	Master	Map the transaction to the tuple (StreamID, SubstreamID, SubstreamIDValid, SSD)
irq_out_command_queue_sync_ns	Signal	Master	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of SIG_IRQ.
irq_out_command_queue_sync_rl	Signal	Master	Pulsed interrupt output signal for realm CMD_SYNC having a completion signal of SIG_IRQ.
irq_out_command_queue_sync_s	Signal	Master	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
irq_out_event_queue_ns	Signal	Master	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
irq_out_event_queue_rl	Signal	Master	Pulsed interrupt output signal for the realm event queue becoming non-empty.
irq_out_event_queue_s	Signal	Master	Pulsed interrupt output signal for the secure event queue becoming non-empty.
irq_out_gerror_ns	Signal	Master	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signaling an error.
irq_out_gerror_rl	Signal	Master	Pulsed interrupt output signal for realm SMMU_GERROR(N) signaling an error.
irq_out_gerror_s	Signal	Master	Pulsed interrupt output signal for secure SMMU_GERROR(N) signaling an error.
irq_out_gpf_far	Signal	Master	For RME-enabled SMMUs. A new error reported in SMMU_ROOT_GPF_FAR will pulse this interrupt.
irq_out_gpt_cfg_far	Signal	Master	For RME-enabled SMMUs. A new error reported in SMMU_ROOT_GPT_CFG_FAR will pulse this interrupt.

Name	Protocol	Type	Description
irq_out_ns	Signal	Master	Pulsed interrupt output signal combined from all non-secure (non-RAS) interrupts. This delivers a pulse if any of the interrupt pins of the specified world also deliver a pulse. SW will have to poll all the different reasons to see why it was delivered.
irq_out_pmcg_ns_as_value	Value	Master	Non-secure PMCG interrupt value port. Value port representing a set of Performance Monitor Counter Group interrupts. There is an unknown number of PMCGs and so an unknown number of PMCG interrupts. There may not necessarily even be an interrupt per group. We export the interrupt to be generated as a unsigned. The 'pmcg_index' is exported on the top 16 bits and the 'pmcg_counter' (that overflowed) on the bottom 16 bits. Any configured MSI for this group will be the next transaction out of the pvbush_m_tw_msi_qs port. The architecture supports an MSI from a PMCG could come out of a different port (say the TBU manager port that a PMCG might be associated with). However the AEM only supports it coming out of the TCU port (pvbush_m_tw_msi_qs).
irq_out_pmcg_s_as_value	Value	Master	Secure PMCG interrupt value port. Value port representing a set of Performance Monitor Counter Group interrupts. There is an unknown number of PMCGs and so an unknown number of PMCG interrupts. There may not necessarily even be an interrupt per group. We export the interrupt to be generated as a unsigned. The 'pmcg_index' is exported on the top 16 bits and the 'pmcg_counter' (that overflowed) on the bottom 16 bits. Any configured MSI for this group will be the next transaction out of the pvbush_m_tw_msi_qs port. The architecture supports an MSI from a PMCG could come out of a different port (say the TBU manager port that a PMCG might be associated with). However the AEM only supports it coming out of the TCU port (pvbush_m_tw_msi_qs).
irq_out_pri_queue	Signal	Master	Pulsed interrupt output signal for the non-secure PRI queue.
irq_out_pri_queue_rl	Signal	Master	Pulsed interrupt output signal for the realm PRI queue.
irq_out_ras_cri_as_value	Value	Master	RAS Critical error interrupt value port. The number of RAS error record groups is dependent on the configuration. As each error record group could have its own interrupts then we write to this value port the error record group number (which is dependent on the configuration) to simulate an interrupt pulse. We only support modelling edge-triggered interrupts. The value passed is formatted as follows: bits[23:16] the ras error group index (if multiple error groups) (0 atm) bits[15:8] node id bits[7:0] the ras record index in the node.
irq_out_ras_eri_as_value	Value	Master	RAS Error Handling Interrupt value port. The number of RAS error record groups is dependent on the configuration. As each error record group could have its own interrupts then we write to this value port the error record group number (which is dependent on the configuration) to simulate an interrupt pulse. We only support modelling edge-triggered interrupts. The value passed is formatted as follows: bits[23:16] the ras error group index (if multiple error groups) (0 atm) bits[15:8] node id bits[7:0] the ras record index in the node.

Name	Protocol	Type	Description
irq_out_ras_fhi_as_value	Value	Master	RAS Fault Handling Interrupt value port. The number of RAS error record groups is dependent on the configuration. As each error record group could have its own interrupts then we write to this value port the error record group number (which is dependent on the configuration) to simulate an interrupt pulse. We only support modelling edge-triggered interrupts. The value passed is formatted as follows: bits[23:16] the ras error group index (if multiple error groups) (0 atm) bits[15:8] node id bits[7:0] the ras record index in the node.
irq_out_rl	Signal	Master	Pulsed interrupt output signal combined from all realm (non-RAS) interrupts. This delivers a pulse if any of the interrupt pins of the specified world also deliver a pulse. SW will have to poll all the different reasons to see why it was delivered.
irq_out_s	Signal	Master	Pulsed interrupt output signal combined from all secure (non-RAS) interrupts. This delivers a pulse if any of the interrupt pins of the specified world also deliver a pulse. SW will have to poll all the different reasons to see why it was delivered.
logptsz_s	Value	Slave	RME: This is a four bit signal that encodes the region size that a single LOGPT entry covers. The default value of this port is derived from the parameter <code>rme_logpt_entry_covers_log2size_in_bytes</code> which is in a different format to the port. If a valid value is driven then it will be put in the field <code>SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ</code> . The port uses the same encoding as the field. If an invalid value is driven to this pin and <code>legacy_tz_en</code> is low then the model will obey the setting of the parameter <code>out_of_range_logptsz_s</code> . This port is sampled at negedge of <code>reset_in</code> and must be set before the negedge of the reset signal.
legacy_tz_en	Signal	Slave	For an RME-enabled SMMU then tie this high to get non-RME behaviour See also the parameter <code>SMMU_ROOT_IDR0</code> . This port is sampled at negedge of <code>reset_in</code> and must be set before the negedge of the reset signal.
pvbus_control_s	PVBus	Slave	Register subordinate port
pvbus_id_routed_m	PVBus	Master	This is a special "id-routed" port for transmitting ATC invalidates and PRI Responses upstream into the PCIe EndPoints, it is not a normal bus. The FastSim ATC invalidate protocol specifies how to route and deal with this this port. See the parameter <code>output_id_routed_transform</code> It is assumed that the StreamID can uniquely route the transaction if there are multiple PCIe Root Complexes.
pvbus_m[64]	PVBus	Master	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered <code>pvbus_s[]</code> port.
pvbus_m_tw_msi_qs	PVBus	Master	Manager port used for Table Walks, MSIs and Queue access when <code>separate_tw_msi_qs_port=true</code>
pvbus_s[64]	PVBus	Slave	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered <code>pvbus_m[]</code> port.
reset_in	Signal	Slave	Reset signal
sev_out	Signal	Master	Event signal

## Parameters for SMMUv3AEM

### **PRESET\_REL\_base\_address**

#### **Type**

uint64\_t

#### **Default value**

0

If using preset addresses (SMMU\_IDR1.QUEUES\_PRESET/TABLES\_PRESET) then the queue and table base registers become fixed. If SMMU\_IDR1.REL then the addresses are relative to the base of the register file and this parameter tells the model what address to add to the queue/table addresses to calculate the actual address.

This is for 'embedded implementations' where the memory for these structures is held within the SMMU itself or in a 'close' RAM. The model does not contain any RAM and the integrator must supply a RAM at the appropriate address.

If the preset tables/queues overlap, the RAM has to implement separate secure and non-secure address spaces.

See also: \* TABLES\_PRESET\_smmu\_{s,r}strtab\_base \*  
TABLES\_PRESET\_smmu\_{s,r}strtab\_base\_cfg \* QUEUES\_PRESET\_smmu\_{s,r}cmdq\_base \*  
QUEUES\_PRESET\_smmu\_{s,r}eventq\_base \* QUEUES\_PRESET\_smmu\_{r}priq\_base (no secure PRIQ)

### **QUEUES\_PRESET\_smmu\_cmdq\_base**

#### **Type**

uint64\_t

#### **Default value**

0

If SMMU\_IDR1.QUEUES\_PRESET == 1 then this is the value that appears in SMMU\_CMDQ\_BASE and SMMU\_CMDQ\_BASE becomes read-only.

See also parameter PRESET\_REL\_base\_address.

### **QUEUES\_PRESET\_smmu\_eventq\_base**

#### **Type**

uint64\_t

#### **Default value**

0

If SMMU\_IDR1.QUEUES\_PRESET == 1 then this is the value that appears in SMMU\_EVENTQ\_BASE and SMMU\_EVENTQ\_BASE becomes read-only.

See also parameter PRESET\_REL\_base\_address.

#### **QUEUES\_PRESET\_smmu\_priq\_base**

##### **Type**

uint64\_t

##### **Default value**

0

If SMMU\_IDR1.QUEUES\_PRESET == 1 and SMMU\_IDR0.PRI == 1 then this is the value that appears in SMMU\_PRIQ\_BASE and SMMU\_PRIQ\_BASE becomes read-only.

See also parameter PRESET\_REL\_base\_address.

#### **QUEUES\_PRESET\_smmu\_r\_cmdq\_base**

##### **Type**

uint64\_t

##### **Default value**

0

If SMMU\_IDR1.QUEUES\_PRESET == 1 then this is the value that appears in SMMU\_R\_CMDQ\_BASE and SMMU\_R\_CMDQ\_BASE becomes read-only.

See also parameter PRESET\_REL\_base\_address.

#### **QUEUES\_PRESET\_smmu\_r\_eventq\_base**

##### **Type**

uint64\_t

##### **Default value**

0

If SMMU\_IDR1.QUEUES\_PRESET == 1 and SMMU\_ROOT\_IDR0.REALM\_IMPL == 1 then this is the value that appears in SMMU\_R\_EVENTQ\_BASE and SMMU\_R\_EVENTQ\_BASE becomes read-only.

See also parameter PRESET\_REL\_base\_address.

#### **QUEUES\_PRESET\_smmu\_r\_priq\_base**

##### **Type**

uint64\_t

##### **Default value**

0

If `SMMU_IDR1.QUEUES_PRESET == 1` and `SMMU_IDR0.PRI == 1` and `SMMU_ROOT_IDR0.REALM_IMPL == 1` then this is the value that appears in `SMMU_PRIQ_BASE` and `SMMU_PRIQ_BASE` becomes read-only.

See also parameter `PRESET_REL_base_address`.

#### **`QUEUES_PRESET_smmu_s_cmdq_base`**

##### **Type**

`uint64_t`

##### **Default value**

0

If `SMMU_IDR1.QUEUES_PRESET == 1` and `SMMU_S_IDR1.SECURE_IMPL == 1` then this is the value that appears in `SMMU_S_CMDQ_BASE` and `SMMU_S_CMDQ_BASE` becomes read-only.

See also parameter `PRESET_REL_base_address`.

#### **`QUEUES_PRESET_smmu_s_eventq_base`**

##### **Type**

`uint64_t`

##### **Default value**

0

If `SMMU_IDR1.QUEUES_PRESET == 1` and `SMMU_S_IDR1.SECURE_IMPL == 1` then this is the value that appears in `SMMU_S_EVENTQ_BASE` and `SMMU_S_EVENTQ_BASE` becomes read-only.

See also parameter `PRESET_REL_base_address`.

#### **`SMMU_AIDR`**

##### **Type**

`uint32_t`

##### **Default value**

0

`SMMU_AIDR` contains the Major and Minor architectural revisions numbers

#### **`SMMU_IDR0`**

##### **Type**

`uint32_t`

##### **Default value**

-

`SMMU_IDR0`. The following fields are further combined with the port `conf_system_supports_{sev,httu,btm,cohacc}`:- `* sev * ht tu * btm * cohacc`

NOTE that SMMU\_IDR0.RME\_IMPL is the value that the SMMU should have if the SMMU is currently RME-aware. It will be forced to zero if the SMMU has been forced to be unaware of RME by legacy\_tz\_en.

**SMMU\_IDR1****Type**

uint32\_t

**Default value**

-

SMMU\_IDR1.

**SMMU\_IDR2****Type**

uint32\_t

**Default value**

0

SMMU\_IDR2 holds the BA\_VATOS field.

**SMMU\_IDR3****Type**

uint32\_t

**Default value**

0

SMMU\_IDR3 is reserved.

**SMMU\_IDR4****Type**

uint32\_t

**Default value**

0

SMMU\_IDR4 is Imp def.

**SMMU\_IDR5****Type**

uint32\_t

**Default value**

0

SMMU\_IDR5 contains, amongst others the output address encoded size (OAS).



**SMMU\_IDR6****Type**

uint32\_t

**Default value**

0

SMMU\_IDR6 is **RES0** if Enhanced Command Queues do not exist (SMMU\_IDR1.ECMDQ == 0).

Otherwise, SMMU\_IDR6 contains information about the configuration of the ECMDQs.

**SMMU\_IIDR****Type**

uint32\_t

**Default value**

0

SMMU\_IIDR contains fields for the implementer, product revision, etc.

**SMMU\_MPAMIDR****Type**

uint32\_t

**Default value**

0

SMMUv3.2: If SMMU\_IDR3.MPAM == 1 then SMMU\_MPAMIDR holds further ID information for Memory Partitioning And Monitoring (MPAM) extension.

This is optional in SMMUv3.2 and is backported to SMMUv3.1.

**SMMU\_ROOT\_IDR0****Type**

uint32\_t

**Default value**

0

If SMMU\_ROOT\_IDR0 is 0 then the SMMU is RME-unaware.

Otherwise...

legacy\_tz\_en is a pin that when high disables RME and the SMMU\_ROOT\_IDR0 register reads as zero.

The effective value of legacy\_tz\_en is derived from

- the last signalled value sampled at negedge of reset

- or if never signalled, the inverse of ROOT\_IMPL (bit[0]) of this parameter.

Thus, ROOT\_IMPL should be zero if we want `legacy_tz_en` to start as high regardless of the actual configuration we want in the SMMU\_ROOT\_IDR0 register when the SMMU is RME-aware.

In other words, if the SMMU is to be RME-aware, then all parameters should be configured as though the SMMU is currently RME-aware with the exception that SMMU\_ROOT\_IDR0.ROOT\_IMPL is the inverse of the default value of `legacy_tz_en`.

SMMU\_ROOT\_IDR0.BGPTM is the default value of the pin `conf_system_supports_bgptm`.

#### **SMMU\_ROOT\_IIDR**

##### **Type**

uint32\_t

##### **Default value**

0

The value of the SMMU\_ROOT\_IIDR register. If is zero then will be the same as SMMU\_IIDR.

#### **SMMU\_R\_AIDR**

##### **Type**

uint32\_t

##### **Default value**

0

The value of SMMU\_R\_AIDR.

#### **SMMU\_R\_IDR0**

##### **Type**

uint32\_t

##### **Default value**

0

The value of SMMU\_R\_IDR0.

#### **SMMU\_R\_IDR3**

##### **Type**

uint32\_t

##### **Default value**

0

The value of SMMU\_R\_IDR3.

**SMMU\_R\_IDR6****Type**

uint32\_t

**Default value**

0

The value of SMMU\_R\_IDR6 that configures the ECMDQs.

**SMMU\_R\_MECIDR****Type**

uint32\_t

**Default value**

0

The value of SMMU\_R\_MECIDR.

**SMMU\_R\_MPAMIDR****Type**

uint64\_t

**Default value**

0xFFFFFFFFFFFFFFFF

NOTE this parameter is 64 bits but the ID register is 32 bits.

This parameter is the value of SMMU\_R\_MPAMIDR, or if ~0ull then it will have the following default values:

- PARTID\_MAX/PMG\_MAX from the SMMU\_MPAMIDR
- HAS\_MPAM\_NS from the SMMU\_S\_MPAMIDR if it exists, otherwise 0.

**SMMU\_S\_IDR0****Type**

uint32\_t

**Default value**

0

Secure IDR0 register.

**SMMU\_S\_IDR1****Type**

uint32\_t

**Default value**

-

SMMU\_S\_IDR1 Indicates if there is a secure side by bit 31.

#### **SMMU\_S\_IDR2**

##### **Type**

uint32\_t

##### **Default value**

0

SMMU\_S\_IDR2 Reserved

#### **SMMU\_S\_IDR3**

##### **Type**

uint32\_t

##### **Default value**

0

SMMU\_S\_IDR3 Reserved

#### **SMMU\_S\_IDR4**

##### **Type**

uint32\_t

##### **Default value**

0

SMMU\_S\_IDR4 IMP DEF

#### **SMMU\_S\_IDR6**

##### **Type**

uint32\_t

##### **Default value**

0

SMMU\_S\_IDR6 is **RES0** if Secure Enhanced Command Queues do not exist (SMMU\_S\_IDR0.ECMDQ == 0).

Otherwise, SMMU\_S\_IDR6 contains information about the configuration of the ECMDQs.

#### **SMMU\_S\_MPAMIDR**

##### **Type**

uint32\_t

##### **Default value**

0

SMMUv3.2: If SMMU\_IDR3.MPAM == 1 then SMMU\_S\_MPAMIDR holds further ID information for Memory Partitioning And Monitoring (MPAM) extension.

This is optional in SMMUv3.2 and is backported to SMMUv3.1.

#### **TABLES\_PRESET\_smmu\_r\_strtab\_base**

##### **Type**

uint64\_t

##### **Default value**

0

If SMMU\_IDR1.TABLES\_PRESET == 1 and SMMU\_ROOT\_IDR0.REALM\_IMPL == 1 then this is the value that appears in SMMU\_R\_STRTAB\_BASE and SMMU\_R\_STRTAB\_BASE becomes read-only.

See also parameter PRESET\_REL\_base\_address.

#### **TABLES\_PRESET\_smmu\_r\_strtab\_base\_cfg**

##### **Type**

uint32\_t

##### **Default value**

0

If SMMU\_IDR1.TABLES\_PRESET == 1 and SMMU\_ROOT\_IDR0.REALM\_IMPL == 1 then this is the value that appears in SMMU\_R\_STRTAB\_BASE\_CFG and SMMU\_R\_STRTAB\_BASE\_CFG becomes read-only.

See also parameter PRESET\_REL\_base\_address.

#### **TABLES\_PRESET\_smmu\_s\_strtab\_base**

##### **Type**

uint64\_t

##### **Default value**

0

If SMMU\_IDR1.TABLES\_PRESET == 1 and SMMU\_S\_IDR1.SECURE\_IMPL == 1 then this is the value that appears in SMMU\_S\_STRTAB\_BASE and SMMU\_S\_STRTAB\_BASE becomes read-only.

See also parameter PRESET\_REL\_base\_address.

#### **TABLES\_PRESET\_smmu\_s\_strtab\_base\_cfg**

##### **Type**

uint32\_t

##### **Default value**

0

If `SMMU_IDR1.TABLES_PRESET == 1` and `SMMU_S_IDR1.SECURE_IMPL == 1` then this is the value that appears in `SMMU_S_STRTAB_BASE_CFG` and `SMMU_S_STRTAB_BASE_CFG` becomes read-only.

See also parameter `PRESET_REL_base_address`.

#### **`TABLES_PRESET_smmu_strtab_base`**

##### **Type**

`uint64_t`

##### **Default value**

0

If `SMMU_IDR1.TABLES_PRESET == 1` then this is the value that appears in `SMMU_STRTAB_BASE` and `SMMU_STRTAB_BASE` becomes read-only.

See also parameter `PRESET_REL_base_address`.

#### **`TABLES_PRESET_smmu_strtab_base_cfg`**

##### **Type**

`uint32_t`

##### **Default value**

0

If `SMMU_IDR1.TABLES_PRESET == 1` then this is the value that appears in `SMMU_STRTAB_BASE_CFG` and `SMMU_STRTAB_BASE_CFG` becomes read-only.

See also parameter `PRESET_REL_base_address`.

#### **`all_error_messages_through_trace`**

##### **Type**

`bool`

##### **Default value**

false

Some conditions in the SMMU are so strange that the software programming the SMMU has done something wrong. At this point messages are output to either `ArchMsg.Error.*` or `ArchMsg.Warning.*` or to the error stream of the simulator. Outputting to the error stream of the simulator may cause it to return with a non-zero exit status.

If you set this option to true then instead of using the error stream of the simulator it will always use a trace stream allowing the simulation to exit with a zero exit status.

#### **`allow_non_secure_access_to_SMMU_S_INIT`**

##### **Type**

`bool`

**Default value**

false

If the system has no software operating as a secure agent then set this parameter. This allows non-secure accesses to the SMMU\_S\_INIT register and allows the non-secure software to reset the TLB, clearing out any 'secure' TLB entries.

If the SMMU does not implement the security extensions (SMMU\_S\_IDR1.SECURE\_IMPL == 0) then this parameter is ignored.

**apply\_ste\_instcfg\_privcfg\_on\_all\_ats\_translated\_accesses****Type**

bool

**Default value**

false

This parameter is ignored if any of the following are true:

- SMMU\_IDR1.ATTR\_PERMS\_OVR == 0
- SMMU\_IDR3.PASIDTT == 1 and the transaction has a PASID

Otherwise, if this parameter is:

- false: STE.INSTCFG/PRIVCFG will only be applied to ATS-TranslatedTransactions if STE.EATS==split-stage.
- true: STE.INSTCFG/PRIVCFG will be applied to all ATS-TranslatedTransactions regardless of the value of STE.EATS.

**ats\_split\_stage\_dbm\_update\_do\_with\_ATSRequest****Type**

unsigned

**Default value**

0

When doing split-stage ATS, then the DBM update for the final stage 2 descriptor can be done either whilst processing the ATS request or delayed until it actually sees the PCIe Translated Transaction using the stage 2 descriptor.

**0**

do when the actual transaction is seen

**1**

do when processing the ATS request

**2**

do it randomly with 50% chance.

**axi\_stream\_msi\_TDEST****Type**

uint32\_t

**Default value**

0

ID of the AXI stream subordinate port on the GIC that receives SMMU originated MSIs sent directly. The name of the GIC port is `axi_stream_msi_s`.

**Note**

If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TID`

**axi\_stream\_msi\_TID****Type**

uint32\_t

**Default value**

0

ID of the AXI stream manager port that sends SMMU TCU originated MSIs directly to the GIC. The name of the SMMU port is `axi_stream_msi_m`.

**Note**

If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TDEST`

**axi\_stream\_msi\_addr\_to\_match****Type**

uint64\_t

**Default value**

0xFFFFFFFFFFFFFFFF



If the last two bits are 0, any SMMU-originated MSI which exactly matches the address will be sent through the AXI stream port `axi_stream_msi_m` which is usually connected to the GIC.

This parameter drives the value of the `axi_stream_msi_addr_to_match_s` port at simulation reset. For every reset after that, the value of the port will be sampled and used if changed.



Note

The entire address must match, including bits [1:0]. As MSIs are 32 bit aligned then if `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

See also:

- `axi_stream_msi_TID`
- `axi_stream_msi_TDEST`

### **behaviour\_of\_sampled\_at\_reset\_signals**

#### **Type**

unsigned

#### **Default value**

0

Some configuration signals into the SMMU are sampled on negedge of reset.

However, it can sometimes be hard to arrange to drive a configuration pin before the negedge of reset.

The configuration pins are sampled:

0

at negedge reset.

1

at negedge reset, but if a later change occurs at the same simulated time, and no transactions have occurred, then they will be resampled and the SMMU reset again.

### **cmdq\_max\_number\_of\_commands\_to\_buffer**

#### **Type**

uint32\_t

#### **Default value**

10

The command queues can buffer fetched commands before issuing them. This parameter is roughly the maximum number of commands to do this for. The programmer visible effects are that just because the CONS pointer shows a command has been consumed does not necessarily mean that it has been issued (and completed). Higher values accentuate this effect.

**dpt\_configure\_ATS\_formed\_entries****Type**

string

**Default value**

"all\_enabled\_stages, vmsa\_and\_gpc"

A comma-separated list of options.

If the string is "" / "none", then no ATS-formed DPT TLB entries are used.

Otherwise...

One of:

- "all\_enabled\_stages" / "" – use all enabled VMSA stages (default)
- "last\_enabled\_stage" – use only the last enabled VMSA stage.

The DPT entry size, one of:

- "vmsa\_and\_gpc" / "" – store as the smaller of the VMSA and GPC region (default)
- "vmsa\_only" – store as the VMSA region

ATS-formed entries and DPT-formed entries can be distinguishable, one of:

- "on\_fault\_always\_walk" / "" – ATS-formed and DPT-formed entries are not distinguished (default)
- "on\_fault\_walk\_if\_ATS\_formed" – ATS-formed entries are not definitive and will cause a walk. DPT-formed entries are definitive and will not cause a walk.

An ATS-formed entry can still be inserted if the only reason it fails is because of the final GPC check, one of:

- "only\_if\_passes\_final\_PA\_GPC\_check"
- "even\_if\_fails\_final\_PA\_GPC\_check"

Example:

"all\_enabled\_stages, vmsa\_and\_gpc"

If we find a DPT TLB entry that is ATS-formed or the implementation does not distinguished ATS-formed or DPT-formed entries then:

- for downgradeable transactions, then if the entry does not allow the transaction but would allow the downgraded transaction:
  - "do\_not\_prefer\_downgrade\_over\_DPT\_walk" – walk the DPT to see if the non-downgraded transaction would be allowed.
  - "prefer\_downgrade\_over\_DPT\_walk" (default) – just do the downgrade and avoid a DPT walk.

- for a NOPpable transaction, then if the entry does not allow the transaction:
  - “do\_not\_prefer\_found\_entry\_NOP\_over\_DPT\_walk” – walk the DPT to see if the original transaction is allowed.
  - “prefer\_found\_entry\_NOP\_over\_DPT\_walk” (default) – just **NOP** the transaction and avoid a DPT walk.

NOTE that if no entry is found then a NOPpable transaction will still perform a walk.

### **dpt\_configure\_invalidation**

#### **Type**

string

#### **Default value**

“”

Configure when entries are invalidated when performing a DPT check.

A comma-separated list of options.

Choose one of:

- lookup\_fault\_invalidates\_any\_existing\_entries / “” (default)
- lookup\_fault\_leaves\_any\_existing\_entries

Choose one of:

- noaccess\_fault\_invalidates\_any\_existing\_entries / “” (default)
- noaccess\_fault\_leaves\_any\_existing\_entries

### **enable\_device\_id\_checks**

#### **Type**

bool

#### **Default value**

true

If this parameter is true then the DeviceIDs seen by the GIC are:

- **for client devices**

`DeviceID = StreamID + translated_device_id_base`

- **for SMMU-generated MSIs**

`smmu_msi_device_id`

This parameter enables two checks:

- If the DeviceID is used in the `output_attribute_transform` parameter, if it overflows 32 bits then the model warns. If the DeviceID is not used, it is assumed that the external agent that forms the DeviceID warns if it overflows.

- If the SMMU supports MSIs, the model checks that the GIC is able to distinguish an MSI generated by the SMMU from one generated by a client device.

As the exact mechanism to determine the DeviceID is in the system and not necessarily under control of the SMMU then you can disable these warnings using this parameter.

See also the parameters: `output_attribute_transform` and `msi_attribute_transform`.

### **`hide_warning_ACCESSEN_GPCEN_set_to_1_in_a_single_write`**

#### **Type**

bool

#### **Default value**

false

The architecture recommends against setting `SMMU_ROOT_CR0.ACCESSEN` and `SMMU_ROOT_CR0.GPCEN` to 1 in the same write if it is possible that a concurrent client device transaction could appear at the SMMU. This is because there could be a time window where the client device transaction sees effective values `ACCESSEN == 1` and `GPCEN == 0` and so would bypass GPC checking.

If your system cannot generate this possibility then you can set this parameter to true and turn off the warning that software has set both `ACCESSEN` and `GPCEN` to 1 at the same time.

### **`hide_warning_EOPD_differs_from_what_would_be_cached`**

#### **Type**

bool

#### **Default value**

false

When this parameter is set to true, warnings that the effective EOPD value differs from what would be cached in the TLB are disabled. False (warnings are showed) by default.

### **`hide_warning_NoStreamID_transaction_for_unsupported_PAS_or_MPAM_SP`**

#### **Type**

bool

#### **Default value**

false

When RME is not supported then a NoStreamID transaction with `PAS[1] == 1` or `MPAM_SP[1] == 1` is treated as though `PAS[1] == 0` and `MPAM_SP[1] == 0`. This is usually a system construction error and is not expected to occur.

The SMMU warns when this occurs, but the warning can be hidden by setting this parameter.

## howto\_identify

### Type

string

### Default value

“use-identify”

If `use-identify` then the SMMU uses the `identify` port to determine the `ssid`, `streamID`, `SubstreamID`. Otherwise, this string extracts them from the transaction’s attributes.

Examples:

```
SEC_SID=ExtendedID[63], SSV=ExtendedID[62], SubstreamID=ExtendedID[51:32],  
StreamID=ExtendedID[31:0]
```

or

```
nSEC_SID=ExtendedID[63], StreamID=ExtendedID[55:24], nSSV=ExtendedID[20],  
SubstreamID=ExtendedID[19:0]  
StreamID[31:24]=0, StreamID[23:0]=ExtendedID[23:0], SSV=1[0], ...
```



Note

If you are not using the `use-identify` option then the configuration string is parsed according to the rules in the section ‘Parameters for parsing transaction attributes’.

LHS Symbols:

#### **SIDV**

Indicates that the StreamID is valid.

#### **SSV**

Indicates that the SubstreamID is valid.

#### **SEC\_SID / SEC\_SID\_bit\_1**

Bits 0 and 1 respectively of the StreamID Security State

#### **StreamID**

(32 b) valid if `sidv` is 1 or both `sidv` and `nsidv` are unused.

#### **SubstreamID**

(20 b) is valid if `ssv` is true.

#### **nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

RHS Symbols:

#### **ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming PVBUS transaction attributes

`SEC_SID` is 1b in the model. For RME systems, in hardware, then `SEC_SID` is 2b, in the model `SEC_SID_bit_1` represents bit[1].

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

Negative and positive logic symbols for the same attribute is an error.

The model uses the term SSD (Security State Determination) to mean which security state the transaction, register, structure, event, etc. belongs to.

In the SMMUv3 Architecture, the side-band signal `SEC_SID` holds the information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one bit signal. With RME, in the hardware, it was expanded to 2b. To retain backwards compatibility in the model, `SEC_SID` remains as 1b in this parameter, but the second bit can be expressed with `SEC_SID_bit_1` (and its negative logic version `nSEC_SID_bit_1`)

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

For those systems that support NoStreamID transactions, and `howto_identify` is not using the port, first the SMMU determines the value of `SIDV` or `nSIDV` to see if the transaction is a NoStreamID transaction (`SIDV == 0` or `nSIDV == 1`). If so then the rest of the `howto_identify` string is ignored as the information extracted relates only to StreamID transactions. Instead more information is extracted using the parameter `howto_identify_NoStreamID_extra_info`.

In AMBA systems, the equivalent signal to `SIDV` is called either `ARMMUVALID` OR `AWMMUVALID`. Collectively they are known as `AxMMUVALID`.

**howto\_identify\_NoStreamID\_extra\_info**

**Type**

string

**Default value**

""

The behavior of this parameter depends on `howto_identify`

- if it equals 'use-identify' then this must be "", otherwise there is an error.
- if it identifies a NoStreamID transaction (`SIDV=0`) then this parameter includes one or more of
  - `MPAM_SP`
  - `MPAM_PARTID`
  - `MPAM_PMG`

- MECID
- HWATTR\_KIND\_0
- in any other case, this parameter is ignored.

Fields set in this parameter must not overlap the `sidv/nsidv` fields in `howto_identify`

Example:

```
MPAM_PMG[7:0]=ExtendedID[62:55], MPAM_PARTID[15:0]=ExtendedID[54:39],
MPAM_SP[1:0]=ExtendedID[38:37], MECID[15:0]=UserFlags[31:16]
HWATTR_KIND_0[3:0]=ExtendedID[42:39]
```



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

## **httu\_early\_st2\_permission\_fault\_if\_af\_update\_at\_stage1**

### **Type**

unsigned

### **Default value**

0

If a stage 1 descriptor needs an HTTU update, but the descriptor is unwriteable at stage 2 and also a stage 1 permission fault occurs, then the architecture permits either the stage 1 or stage 2 permission fault to be recorded.

0 – stage 1 permission check check stage 1 descriptor writeable at stage 2 if AF- or DBM-update required

1 – check stage 1 descriptor writeable at stage 2 if AF-update required stage 1 permission check check stage 1 descriptor writeable at stage 2 if DBM-update required

2 – do behaviour 1 or 2 randomly with a 50% chance.

## **httu\_memory\_types\_supported**

### **Type**

string

### **Default value**

“rawaWB, raWB, waWB, naWB”

This is a comma-separated list of memory types that are **IMPLEMENTATION DEFINED** as supporting HTTU. However, the system must have Far Atomic support for the specified memory address and memory type.

Device types:

- nGnRnE, nGnRE, nGRE, GRE

Normal memory types are composed of an ‘inner’ and an ‘outer’ cacheability. The model only supports types where the inner and outer are identical.

- Normal non-cacheable types
  - nc\_nb, nc
- Cacheable types are of the form (na?|(ra)?(wa)?)(WT|WB)(tr)?
  - na/ra/wa – no/read/write allocate
  - WT/WB – write through/write back
  - tr – transient
  - exceptions:
    - ‘na’ and ‘tr’ are incompatible
    - without ‘na’ then you must specify at least one of ra/wa. Example: “WT” is illegal, “raWT” is legal.

rawaWB is always supported and it is optional.

Examples:

- “rawaWB, raWB, waWB, naWB” – only the WB type is supported
- “nc” – rawaWB and the normal non-cacheable type are supported

### **imp\_def\_L1CD\_L2Ptr\_out\_of\_range**

#### **Type**

unsigned

#### **Default value**

0

If an L1CD.L2Ptr is out of range of IAS/OAS as appropriate then what happens is controlled by this parameter: 0 – if is an IPA, then Stage 2 Translation Fault, if is a PA then truncate to OAS. 1 – generate C\_BAD\_SUBSTREAMID if an IPA and > IAS, or if a PA and > OAS. 2 – generate C\_BAD\_SUBSTREAMID if an IPA and > IAS, or F\_CD\_FETCH if a PA and > OAS. 3 – truncate the IPA or PA to IAS/OAS as appropriate

NOTE that if the model is configured as SMMUv3.1 then this parameter is IGNORED, and behaves as though this parameter was set to 1. The SMMUv3.1 architecture actually allows more behaviours but the model will only implement this one.

NOTE that the SMMUv3.0 allows more behaviours than can be expressed by this parameter.



**imp\_def\_PID0****Type**

unsigned

**Default value**

0x83

If imp\_def\_has\_PID\_CID is true then this is the PID0 value.

**imp\_def\_PID1****Type**

unsigned

**Default value**

0xb4

If imp\_def\_has\_PID\_CID is true then this is the PID1 value.

**imp\_def\_PID2****Type**

unsigned

**Default value**

0xb

If imp\_def\_has\_PID\_CID is true then this is the PID2 value.

**imp\_def\_PID3****Type**

unsigned

**Default value**

0x0

If imp\_def\_has\_PID\_CID is true then this is the PID3 value.

**imp\_def\_PID4****Type**

unsigned

**Default value**

0x4

If imp\_def\_has\_PID\_CID is true then this is the PID4 value.

**imp\_def\_S1ContextPtr\_out\_of\_range****Type**

unsigned

**Default value**

0

If an STE is fetched that uses a stage 1 then if: - Stage 1 only and S1ContextPtr > OAS, or - Stage 1+2 and S1ContextPtr > IAS then what happens is IMP DEF and this parameter controls the behaviour:- 0 – stage 1 only – C\_BAD\_STE – stage 1+2 – C\_BAD\_STE 1 – stage 1 only – C\_BAD\_STE – stage 1+2 – truncate to IAS 2 – stage 1 only – truncate to OAS – stage 1+2 – C\_BAD\_STE 3 – stage 1 only – truncate to OAS – stage 1+2 – truncate to IAS 4 – stage 1 only – truncate to OAS – stage 1+2 – Stage 2 translation fault 5 – stage 1 only – C\_BAD\_STE – stage 1+2 – Stage 2 translation fault The architecture also allows for F\_CD\_FETCH, but the model does not support this.

NOTE that in SMMUv3.1 then the only allowed values of this parameter are 0 or 5.

**imp\_def\_alloccfg****Type**

unsigned

**Default value**

0

ALLOCCFG overrides the read/write/transient hints on cacheable types. However these are hints and an implementation may choose to treat them differently.

**0**

apply the alloc hints as architecturally specified

**1**

ignore all ALLOCCFG fields (treated as zero)

**2**

apply ALLOCCFG only when MTCFG == 1

**imp\_def\_apply\_dre\_dcp\_to\_full\_ats****Type**

bool

**Default value**

false

STE.DRE and STE.DCP control the downgrade of certain transactions to **NOP**.

If this parameter is true, then for ATS-TranslatedTransactions using STE.EATS == all\_stages or use\_dpt then the STE.DRE/STE.DCP controls are applied.

For ATS-TranslatedTransactions using STE.EATS == eats\_use\_dpt then the controls are always applied.

For SMMUv3.4 and later, or if SMMU\_IDR3.DPT == 1 then this parameter is ignored and treated as 'true'.

### **imp\_def\_ats\_attribute\_stashing**

#### **Type**

unsigned

#### **Default value**

0

The SMMU architecture allows an ATS request to return the attributes with which to make the Translated Access. PCIe does not define any transaction attributes in the ARM sense and so the mechanism for doing this is IMP DEF. Usually this would be done by packing them into the high order address bits of the return response.

In the model, then the representation of the ATS reply returns the attributes directly and it is up to the ATC whether it wants to use them or not.

The parameter configures what to place in those architectural attributes in the ATS Reply.

**0**

the architectural attributes

**1**

Inner Write Back, Outer Write Back, Inner Shared, read and write allocate, User-Data

**2**

Inner Write Back, Outer Write Back, Outer Shared, read and write allocate, User-Data

The SMMU cannot force an ATC to use these attributes, it is simply the attributes that are returned in the non-PCIe part of the ATS reply.

### **imp\_def\_ats\_response\_stu**

#### **Type**

unsigned

#### **Default value**

0

A successful ATS Response with RW != 0 can return any-sized region from the STU to the actual region size. The chosen size is:

- 0 – use the full size of the region
- 1 – use the STU
- 2 – use half the size of the region

**imp\_def\_cohacc\_effect****Type**

unsigned

**Default value**

0

SMMU\_IDR0.COACC is a system property. However, the exact nature of the transactions that the SMMU emits is an IMP DEF property when COACC == 0:

**0**

COACC == 0 forces the output attributes of SMMU-generated accesses to non-shared.

**1**

The only effect of COACC is what is reported in SMMU\_IDR0.COACC and has no effect on the output attributes of SMMU-generated accesses.

**imp\_def\_contiguous\_bit\_handling****Type**

unsigned

**Default value**

0

If the Contiguous bit is set in a translation table descriptor then modify how it is cached:

- 0 – use the full size of the contiguous region
- 1 – ignore the contig bit for determining the region size
- 2 – use half the size of the contiguous region

See imp\_def\_gpt\_contiguous\_bit\_handling for GPT Contig bit handling.

**imp\_def\_effective\_ATTR\_TYPES\_OVR\_is\_false\_per\_port****Type**

string

**Default value**

"0"

SMMU\_IDR1.ATTR\_TYPES\_OVR == 1 means that the STE and SMMU\_(S\_)GBPA MTCFG/SHCFG/ALLOCCFG have an effect.

However, an implementation is allowed to ignore this being one for specific ports and *not* apply the overrides MTCFG/SHCFG/ALLOCCFG despite SMMU\_IDR1.ATTR\_TYPES\_OVR == 1.

This parameter is a comma-separated lists of port ranges (indexed from 0) for those ports where SMMU\_IDR1.ATTR\_TYPES\_OVR behaves as 0. For example:

```
0, 10-20, 40
```

### **imp\_def\_gpt\_contiguous\_bit\_handling**

#### **Type**

unsigned

#### **Default value**

0

If we find a GPT Contiguous descriptor then modify how it is cached:

- 0 – use the full size of the contiguous region
- 1 – ignore the contig bit for determining the region size and use PGS
- 2 – use half the size of the contiguous region

See `imp_def_contiguous_bit_handling` for VMSA Contig bit handling.

### **imp\_def\_has\_PID\_CID**

#### **Type**

bool

#### **Default value**

true

If this is true then the SMMU model will have the standard PID/CID ID registers. Only the PID0..PID4 registers can be customized and the parameters `imp_def_PID0..imp_def_PID4` are used.

### **imp\_def\_no\_InD\_PnU\_on\_downstream\_system**

#### **Type**

bool

#### **Default value**

false

SMMUv3.4 deprecated the SMMU emitting the Instruction/Data (InD) and the Privileged/User (PnU) markings onto the downstream interconnect as:

- many interconnects have no way to represent them,
- the downstream system should not care about the value of these fields.

If this parameter is set to true then all transactions are marked as Privileged-Data on the downstream interconnect.

From SMMUv3.4, this parameter is ignored and has the effective value 'true'.

**imp\_def\_ns\_bit\_for\_s\_gatos\_on\_s1\_bypass\_non\_sel2****Type**

uint32\_t

**Default value**

0

This parameter only has an effect if SEL2 == 0.

When SEL2 == 0, Secure virtualisation is not supported and only stage 1 is supported for Secure Streams.

In this case, the Secure ATOS interface does not provide a mechanism to specify the input NS bit to the stage 1 translation. The input bit is IMP DEF and only has an effect if the transaction has no SubstreamID and bypasses by S1DSS.

This parameter specifies the IMP DEF input bit: 0 – secure 1 – non-secure 2 – random

See also `imp_def_ns_bit_for_s_vatos_on_s1_bypass` which configures something similar for the Secure VATOS (not GATOS) interface.

**imp\_def\_ns\_bit\_for\_s\_vatos\_on\_s1\_bypass****Type**

uint32\_t

**Default value**

0

When a Secure VATOS operations for a translation that bypasses stage 1 by S1DSS then the output NS bit is the same as the input NS bit of the translation.

The architecture does not provide an input NS bit in the SMMU\_S\_VATOS\_ADDR register and it is treated as an IMP DEF value.

This parameter specifies that value: 0 – secure 1 – non-secure 2 – random

See also `imp_def_ns_bit_for_s_gatos_on_s1_bypass_non_sel2` which configures something similar for the Secure GATOS (not VATOS) interface.

**imp\_def\_ras\_access\_control\_policy****Type**

string

**Default value**

"use-imp\_def\_ras\_allow\_non\_secure\_accesses\_if\_supports\_secure"

The access control of the RAS nodes:

- "" / "use-imp\_def\_ras\_allow\_non\_secure\_accesses\_if\_supports\_secure"

- The parameter `imp_def_ras_allow_non_secure_accesses_if_supports_secure` is used to determine access
- “non-secure”
  - Allow access to all PAses.
- “secure”
  - Only allow access to secure and root. If secure is not implemented then will allow access to non-secure.
- “root-only”
  - The register file is only accessible to root-pas transactions. If root is not implemented then it will act as though this parameter was: “use-`imp_def_ras_allow_non_secure_accesses_if_supports_secure`”

### **`imp_def_ras_allow_non_secure_accesses_if_supports_secure`**

#### **Type**

bool

#### **Default value**

false

If two security worlds are supported, i.e.: `SMMU_S_IDR1.SECURE_IMPL == 1` then if this parameter is true, then non-secure accesses are allowed to access any RAS registers (see parameter ‘ras’). Otherwise, non-secure accesses are **RAZ/WI**.

If only a single security state (non-secure) is supported, then this parameter is ignored and non-secure accesses are always allowed.

See also `imp_def_ras_access_control_policy` that can override this parameter.

### **`imp_def_reset_unknown_fields_to_zero`**

#### **Type**

bool

#### **Default value**

false

Many fields and registers in the SMMUv3 architecture reset to an **UNKNOWN** value. However, many implementations will choose to reset to 0. By setting this parameter to true then those fields will be initialised to zero.

### **`imp_def_rme_gpf_syndrome_for_PMCg_MSIs`**

#### **Type**

string

#### **Default value**

“other\_gpf”

An MSI access from a PMCG that experiences a GPF is permitted to be reported as either of:

- REASON = GERROR and FAULTCODE = OTHER\_GPF
- REASON = TRANSACTION

The values of this string are one of: \* other\_gpf \* transaction

See also the parameter `imp_def_rme_gpf_syndrome_for_RAS_MSIs`

#### **`imp_def_rme_gpf_syndrome_for_RAS_MSIs`**

##### **Type**

string

##### **Default value**

"other\_gpf"

An MSI access from a RAS record interrupt that experiences a GPF is permitted to be reported as either of:

- REASON = GERROR and FAULTCODE = OTHER\_GPF
- REASON = TRANSACTION

The values of this string are one of: \* other\_gpf \* transaction

See also the parameter `imp_def_rme_gpf_syndrome_for_PMCG_MSIs`

#### **`imp_def_rme_mpam_info_from_NoStreamID_on_gpt_walks`**

##### **Type**

string

##### **Default value**

""

The MPAM related fields in `howto_identify_NoStreamID_extra_info` can be used on the GPT walks for NoStreamID transactions.

If empty then `imp_def_rme_mpam_info_from_NoStreamID_on_gpt_walks_ignored` is obeyed.

If non-empty then is a comma-separated list.

One of:

- `mpam_sp:pas` – the PAS is used as the MPAM\_SP
- `mpam_sp:incoming` or "" – the incoming MPAM\_SP is used, or 0 if it doesn't have one.

One of:

- `partid/pmg:0` – the PARTID/PMG is 0



- `partid/pmg:incoming` – the PARTID/PMG is the untruncated incoming, or 0 if it doesn't have one.
- `partid/pmg:truncate_or_0` or `""` – same as `partid/pmg:incoming` but truncated to the appropriate `SMMU_MPAMIDR` / `SMMU_S_MPAMIDR` or `SMMU_R_MPAMIDR`. If the register does not exist, use 0.
- `partid/pmg:truncate` – same as `partid/pmg:truncate_or_0` but if the register does not exist, use the maximum bit width of implied by all of registers that do exist.

See also: `imp_def_rme_mpam_info_on_NoStreamID`.

### **`imp_def_rme_mpam_info_from_NoStreamID_on_gpt_walks_ignored`**

#### **Type**

bool

#### **Default value**

false

The MPAM related fields set in `howto_identify_NoStreamID_extra_info` are ignored when this parameter is set. This parameter only makes sense when `howto_identify` equals `use-identify` so in any other case it must be false.

When this parameter is set the MPAM information for GPT walks is:

- `MPAM_SP` = PAS
- `MPAM_PARTID` = 0
- `MPAM_PMG` = 0

This parameter is ignored if `imp_def_rme_mpam_info_from_NoStreamID_on_gpt_walks` is not empty.

### **`imp_def_rme_mpam_info_on_NoStreamID`**

#### **Type**

string

#### **Default value**

""

The MPAM related fields in `howto_identify_NoStreamID_extra_info` can be used for the MPAM information on downstream NoStreamID transactions.

If non-empty then is a comma-separated list of options.

One of:

- `mpam_sp:pas` – the PAS is used as the `MPAM_SP`
- `mpam_sp:incoming` or `""` – the incoming `MPAM_SP` is used, or 0 if it doesn't have one.

One of:

- `partid/pmg:0` – the PARTID/PMG is 0
- `partid/pmg:incoming` – the PARTID/PMG is the untruncated incoming, or 0 if it doesn't have one.
- `partid/pmg:truncate_or_0` Or "" – same as `partid/pmg:incoming` but truncated to the appropriate `SMMU_MPAMIDR` / `SMMU_S_MPAMIDR` OR `SMMU_R_MPAMIDR`. If the register does not exist, use 0.
- `partid/pmg:truncate` – same as `partid/pmg:truncate_or_0` but if the register does not exist, use the maximum bit width of implied by all of registers that do exist.

See also: `imp_def_rme_mpam_info_from_NoStreamID_on_gpt_walks`.

### **`imp_def_split_ATS_attributes_is_stage1`**

#### **Type**

bool

#### **Default value**

false

If using split stage ATS, then it is IMP DEF whether the stage 1 attributes are returned to the ATS request or stage 2.

This only has a meaning if the SMMU can stash attributes in the ATS response.

### **`imp_def_truncate_out_of_range_streamids_on_invalidate_commands`**

#### **Type**

bool

#### **Default value**

false

If this parameter is true then the StreamID fields of the following commands will be truncated to (S\_)SIDSIZE:

- `CMD_ATC_INV`
- `CMD_CFGI_STE`
- `CMD_CFGI_STE_RANGE`
- `CMD_CFGI_CD`
- `CMD_CFGI_CD_ALL`

Otherwise, these commands will **NOP**.

### **`imp_def_v3_atos_fault`**

#### **Type**

unsigned

**Default value**

0

For an IPA to PA ATOS translation that encounters a Stage 1 Address Size Fault then the PAR.REASON field reports: \* in SMMUv3.1, 'Stage 1' (0) \* in SMMUv3.0, 'Stage 1' (0) or 'Input' (3) depending on the implementation.

This parameter is ignored for SMMUv3.1.

For SMMUv3.0 then the values are: 0 – report as 'Input' (3) 1 – report as 'Stage 1' (0)

**ish\_is\_osh****Type**

bool

**Default value**

false

When set, any transaction that would use the architectural inner shareable domain is converted to use the outer shareable domain.

NOTE that this parameter should match the equivalent ish\_is\_osh from the PE. If an incompatible value of the ish\_is\_osh parameter is configured for the PE and the SMMU, data coherency may be compromised.

**mec\_attribute\_transform****Type**

string

**Default value**

""

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MEC is supported, this is applied to *all* downstream transactions to transport the MEC information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"UserFlags[31:16]=MECID[15:0]"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/MasterID/UserFlags.

RHS Symbols:

- MECID

Any bits with no transform are unchanged.



Note

Attribute transforms applied before this:

- for client transactions `output_attribute_transform / output_attribute_transform_for_NoStreamID`.
- for table walks `tw_qs_attribute_transform`.
- for MSIs `msi_attribute_transform`.
- if MPAM is enabled `mpam_attribute_transform`.

## **`mpam_attribute_transform`**

### **Type**

string

### **Default value**

"ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_NS"



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MPAM is supported, this is applied to *all* downstream transactions to transport the MPAM information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID,
ExtendedID[38]=MPAM_SP[0]"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/MasterID/UserFlags.

RHS Symbols:

- MPAM\_PARTID
- MPAM\_PMG
- MPAM\_NS
- MPAM\_SP

- numeric literals

Any bits with no transform are unchanged.



- attribute transforms applied before this:
  - for client transactions `output_attribute_transform / output_attribute_transform_for_NoStreamID`.
  - for table walks `tw_qs_attribute_transform`.
  - for MSIs `msi_attribute_transform`.
- `mec_attribute_transform` is applied after this.
- for translated transactions from client devices then `MPAM_NS = ! SEC_SID`.

### **`mpam_sp_options`**

#### **Type**

unsigned

#### **Default value**

2

The width of the MPAM\_SP output side-band information.

The SMMU architecture says that the SMMU is a four-space MPAM component when RME-DA is implemented. However, it can potentially be converted to a two-space MPAM at the edge of the SMMU.

This parameter controls the width of the MPAM\_SP output side-band information:

- 1 – 1b, conventionally the side-band is then called `MPAM_NS`. Any 2b MPAM\_SP value generated will have bit[1] forced to zero.
- 2 – 2b, the side-band is 2b

The same effect can be achieved by using the parameter `mpam_attribute_transform` to only export a single bit of the MPAM\_SP. However, this option allows a model system to be built with a single static version of `mpam_attribute_transform` and then dynamically switch its behaviour more simply.

### **`msi_attribute_transform`**

#### **Type**

string

#### **Default value**

"ExtendedID[31:0]=smmu\_msi\_device\_id, ManagerID64[31:0]=0xFFFFffff"

Transform downstream attributes of MSI transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter output attributes of SMMU-generated MSIs. Example:

```
"UserFlags[15:0]=smmu_msi_device_id[31:16],
ManagerID64[15:0]=smmu_msi_device_id[15:0],
ExtendedID=0"
```

LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Outgoing PVBUS transaction attributes

RHS Symbols:

**smmu\_msi\_device\_id**

The value stored in the parameter with the same name

**interrupt\_kind**

The selected bit corresponds to the interrupts listed below

**0/1**

EVENTQ s/ns

**2**

PRIQ

**3/4**

CMD\_SYNC s/ns

**5/6**

GERROR s/ns

**7/8**

PMCG s/ns

**9/10/11**

RAS FHI/ERI/CRI

**12/13**

gpf\_far/gpt\_cfg\_far

**14/15/16/17**

Realm EVENTQ/PRIQ/CMDQ/GERROR

**HWATTR\_KIND\_0**

PBHA information

**Numeric Literals**

Any number. Ex: 0x1234

ExtendedID/ManagerID64/MasterID/UserFlags start with values {0, 0xFFFFFFFF, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.

This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices.



See also `output_attribute_transform` and `enable_device_id_checks`.



After 11.25 the `interrupt_kind` field was extended to 5 bits. This is strictly a non-backwards compatible change. However, the original 3 bits were insufficient to express all the interrupt kinds that exist.

---

**msi\_ra\_wa\_tr**

**Type**

uint32\_t

**Default value**

7

A bitmap of the Read Allocation, Write Allocate and Transient hints for MSIs to cacheable memory: bit[0] Transient bit[1] Write Allocate bit[2] Read Allocate If not Write Allocate then it will be forced to Read Allocate as a limitation of AMBA.

**non\_arch\_incoming\_stronger\_than\_iWB\_oWB\_forces\_output\_iNC\_oNC\_or\_stronger**

**Type**

string

**Default value**

{}

If not empty, then this enables a specific non-architectural behaviour on the comma-separated list of port indexes, or ranges. For example:

0, 10-20, 40

In the normal translation process, then the input attributes are usually replaced by the attributes from the page tables or SMMU\_(S\_)GBPA.

The behaviour is:

if incoming attributes are iWB-oWB use the architectural attributes else use the stronger of iNC-oNC-osh and the architectural attributes.

This is useful if the ports represent transactions from the PCIe subsystem and the PCIe devices output: \* iWB-oWB if not No\_Snoop -> output is architectural attributes \* iNC-oNC-osh if No\_Snoop -> output is iNC-oNC-osh or stronger

#### **normalize\_input\_normal\_non\_iWB\_oWB\_to\_iNC\_oNC\_osh**

##### **Type**

bool

##### **Default value**

false

When set, use Inner Non Cacheable, Outer Non Cacheable, Outer Shareable for any upstream transaction that would use any of the following attributes: \* Normal Non-cacheable Bufferable \* Normal Non-cacheable Non-bufferable \* Write-through

NOTE that this parameter should match the equivalent configuration from the PE. If an incompatible value of this parameter is configured for the PE and the SMMU, data coherency may be compromised.

#### **number\_of\_ports**

##### **Type**

unsigned

##### **Default value**

1

The number of port pairs that the SMMU has.

#### **nw\_dcp\_extra\_drop\_conditions**

##### **Type**

string

##### **Default value**

""

NW-DCP is a hint and can be dropped for any reason.

This is a comma-separated list of:

- "INST" – NW-DCP is architecturally 'data' but by enabling this option then STE.INSTCFG is applied and it can fault due to SIF and, for rl-ssd, preventing instruction access to ns-PAS.
- "SIF-cached" – NW-DCP needs any of rxw permissions to go downstream. If SIF has been cached into the TLB entry then it will have removed execute permission and so for an execute-only page the NW-DCP would be denied.



If you set this option then, for NW-DCP, the SMMU will behave as if SIF was cached.

The parameter `ordering_of_PAN_and_xn_by_ns_pas` can force TLB-caching of SIF and takes precedence over this option.

- “combined-st2” – when considering the stage 2 permissions then they are first combined with any stage 1 permissions before applying the permission check.
- “combined-st1-st2” – always fetches all stages and combine before applying the permissions check.

### **`ordering_of_PAN_and_xn_by_ns_pas`**

#### **Type**

unsigned

#### **Default value**

0

Execution from ns-pas can be forbidden:

- for secure, this is controlled by `SMMU_S_CR0.SIF`.
- for realm, this is mandatory. In the model, we call this RIF and is always cached in the TLB.

When PAN is interpreted as EPAN then whether SIF/RIF is applied before or after PAN can get different results for the direct permission model.

In the model, the following options are available:

0 – PAN applied first, SIF not cached in the TLB, RIF cached in the TLB. 1 – PAN applied first, SIF/RIF cached in the TLB. 2 – SIF/RIF cached in the TLB, then PAN applied

NOTE that if you cache SIF in the TLB then all SIF faults are no longer traced separately as SIF faults but as permission faults – which architecturally they are reported as. As RIF is always cached in the TLB, then they are not distinguished in the trace separately to permission faults.

### **`out_of_range_CMD_ATC_INV_Size`**

#### **Type**

unsigned

#### **Default value**

0

If `CMD_ATC_INV.Size > 52` then the model is allowed to:- 0 – raise `CERROR_ILL` 1 – treat as **NOP**

The architecture also allows for an **UNKNOWN** invalidate size to be used as well but the model does not support this.

**out\_of\_range\_l0gptsz\_s****Type**

int32\_t

**Default value**

-1

If the port l0gptsz\_s is driven to an invalid value and that value is used then the following behaviors are possible:

- -2 – report the incoming value in SMMU\_ROOT\_GPT\_BASE\_CFG.LOGPTSZ and all transactions will report a GPT Config Error if GPC checking is enabled.
- -1 – produce an error and make the model unusable (default)
- invalid LOGPTSZ encoding – report this value in SMMU\_ROOT\_GPT\_BASE\_CFG.LOGPTSZ and all transactions will report a GPT Config Error if GPC checking is enabled.

All other values are reserved and act as -1.

**output\_attribute\_transform****Type**

string

**Default value**

"ExtendedID[31:0]=DeviceID"

Transform downstream attributes of StreamID transactions.

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV, UserFlags[19:0]=SubstreamID,
ManagerID64[10]=ManagerID64[11], ManagerID64[11]=ManagerID64[10]"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**DeviceID**

StreamID + translated\_device\_id\_base

**StreamID / SubstreamID / SEC\_SID / SEC\_SID\_bit\_1 / SIDV / SSV**

Architectural information. See parameter `howto_identify` for more information.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

**St1PBHA / St2PBHA**

Page Based Hardware Attributes from leaf descriptors (zero if not used).

**STE\_IMPDEF1**

STE[127:116]

**HWATTR\_KIND\_0**

PBHA information in the format of MMU-700 and later

**Numeric Literals**

Any number. Ex: 0x1234

Any bits with no transform are unchanged.



Pre-RME, the SEC\_SID *input* to the SMMUv3 was a single bit. RME added a second bit. In order for the model to be backwards-compatible, the SEC\_SID *symbol* in this parameter remains as a single bit and a new symbol SEC\_SID\_bit\_1 has to be used to refer to the second bit of the SEC\_SID *input*.



- `mpam_attribute_transform` and `mec_attribute_transform` are applied in order after this.
- See also `output_attribute_transform_for_NoStreamID` for NoStreamID transactions.

**output\_attribute\_transform\_for\_NoStreamID**

**Type**

string

**Default value**

“ExtendedID[31:0]=0, ExtendedID[32]=1”



The parameter is parsed according to the rules in the section ‘Parameters for parsing transaction attributes’.

Transform downstream attributes of NoStreamID transactions.

- “” or “none”: no transform

- How to alter output attributes. Example:

```
"ExtendedID[15:0]=0, UserFlags[31]=1, UserFlags[19:0]=0,
ManagerID64[10]=ManagerID64[11],
ManagerID64[11]=ManagerID64[10] ManagerID64[9:6]=HWATTR_KIND_0"
```

RHS/LHS Symbols: \* ExtendedID/ManagerID64/MasterID/UserFlags: incoming/outgoing attributes.

RHS Symbols: \* SIDV = 0, nSIDV = 1 (fixed values to indicate NoStreamID) \* PAS \*  
HWATTR\_KIND\_0

Any bits with no transform are unchanged.



Note

- mpam\_attribute\_transform and mec\_attribute\_transform are applied in order after this.
- see also output\_attribute\_transform for StreamID transactions.

## output\_id\_routed\_transform

### Type

string

### Default value

"Address[43:12]=StreamID, PAS=SSD"



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

The SMMU generates the following ID-routed transaction on the pvbus\_id\_routed\_m bus:

- ATC Invalidate
- PRI Response

This parameter controls how the SMMU should express:

- the StreamID
- the Trusted (T) bit

The value is a comma-separated list of assignments:

```
Address[27:12]=StreamID[15:0], ExtendedID[60]=T, ExtendedID[15:0]=StreamID[31:16]
```

Address bits[11:0] cannot be used.

The LHS can be one of:

- PAS
- MasterID / ManagerID64 / ExtendedID / UserFlags
- Address

The RHS can be one of:

- a numeric constant
- SSD
- T or negative version nT
- StreamID

For realm (or 'Trusted') transactions, then  $ssd=0b11$ ,  $T=1$ ,  $nT=0$ . For non-secure (or 'Non-Trusted') transactions, then  $ssd=0b01$ ,  $T=0$ ,  $nT=1$ .

### **percent\_commit**

#### **Type**

uint32\_t

#### **Default value**

20

Percentage of times that a read of a register with Update will commit the update. 0 means commit immediately.

### **percent\_commit\_Update\_clear**

#### **Type**

uint32\_t

#### **Default value**

20

Percentage of times that a read of a register with a pending Update clear will lower the Update flag.

### **pmu**

#### **Type**

string

#### **Default value**

""

What to instantiate as a PMU.

NOTE that all events and counters are intended for demonstration purposes only and should not be treated as in any way reflecting accurate values for a real implementation. The model's internal representation of actions differ significantly from real hardware and the particular value obtained from the counters should not be used for benchmarking.

Values of this parameter are: \* "" – no PMU \* "distributed-0" \* "distributed-1"

distributed-0: \* a PMCG per TBU (number\_of\_ports, up to 63 ports) \* a single PMCG for a TCU \*  
Connect a debugger to see the configuration.

distributed-1: \* same as distributed-0, except for supporting MSIs and MPAM on the MSIs if  
MPAM is supported by rest of the SMMU.

## **ports\_that\_ignore\_PnU\_InD\_on\_transactions\_with\_no\_SubstreamID**

### **Type**

string

### **Default value**

""

Some bus systems (notably PCIe) do not support marking a transaction as Privileged/User or Instruction/Data unless the transaction has a SubstreamID.

This accepts a comma separated list of numbers and ranges, for example:

0, 10-12, 15

If the number P is named in this list then the upstream pvbus\_s[P] will have all transactions with no Substream considered to be User and Data.

## **prefetch\_only\_requests**

### **Type**

unsigned

### **Default value**

0

The simulator supports 'prefetch-only' DMI requests, which can occur at any time and for any reason and are intended to be invisible to the end execution of the model and to the user.

**0**

deny all prefetch-only requests

**1**

- use debug requests for any page table walks
  - form and use debug TLB/cache entries
  - any faults will not record, but deny the prefetch request

**2**

- treat prefetch-only requests like normal transactions
  - use normal page table walk transactions
  - use and form normal TLB/cache entries

- faults will alter the programmer-visible state of the SMMU

0 is the safest.

1 treats the access like a debug request and requires that debug page table walks are treated correctly downstream. Any descriptors that need HTTU to allow the transaction to proceed will fail the request.

2 is dangerous, it uses real transactions and reports faults that are unphysical. Real transactions can be `wait()`ed and this disobeys the SystemC spec for `get_direct_mem_ptr()`.

## **ras**

### **Type**

string

### **Default value**

""

What to instantiate for RAS handling.

Values of this parameter are: \* "" – no RAS records \* "MMU\_600" \* "MMU\_700" \* "MMU\_S3"

MMU\_600: \* only corrected errors reported.

See also `imp_def_ras_allow_non_secure_accesses_if_supports_secure`.

## **register\_accesses\_to\_root\_or\_realm\_pas\_when\_no\_rme**

### **Type**

uint32\_t

### **Default value**

0

When RME is not implemented or disabled by `legacy_tz_en`:

**0**

root and realm register PAS accesses are not **RAZ/WI**

**1**

root and realm register PAS accesses are treated as **RAZ/WI**

## **reset\_value\_of\_SMMU\_GBPA**

### **Type**

uint32\_t

### **Default value**

0

Reset value of SMMU\_GBPA

**reset\_value\_of\_SMMU\_S\_GBPA****Type**

uint32\_t

**Default value**

0

Reset value of SMMU\_S\_GBPA

**rme\_ats\_request\_pa\_strategy****Type**

uint32\_t

**Default value**

0

When RME\_IMPL == 0, the PA of an ATS Request's response is permitted but not required to undergo a GPT check:

0 – do not check the PA 1 – do the check against the PA 2 – check the PA 50% of the time

Translated transactions are required to always undergo a GPT check whatever happens.

This parameter is ignored if RME\_IMPL==1 and the PA is required to be checked.

**rme\_da\_force\_better\_configuration****Type**

string

**Default value**

""

RME-DA requires that the SMMU be integrated into a system for which SMMU\_IDR0.COHAAC == 1 and SMMU\_IDR0.IDR0.HTTU == both\_af\_and\_dirty (2).

The model has pins:

- conf\_system\_supports\_cohacc
- conf\_system\_supports\_httu

that can control these ID fields.

In addition, RME-DA requires that the fundamental SMMU has certain properties that are configured by its ID codes.

This parameter allows you to selectively ignore the pins and bad ID to produce a good configuration by forcing the required values.

Whether the SMMU has RME-DA or not is identified by SMMU\_ROOT\_IDR0.REALM\_IMPL.



This is a comma-separated list of fields to force when RME-DA is configured by SMMU\_ROOT\_IDR0.REALM\_IMPL:

- in SMMU\_IDR0:
  - “Hyp”
  - “S1P”
  - “S2P”
  - “TTF”
  - “NS1ATS”
  - “COHACC”
  - “HTTU”
  - “RME\_IMPL”
- “SSIDSIZE” in SMMU\_IDR1
- “BBML” in SMMU\_IDR3

You can also use “all” to set all.

### **rme\_10gpt\_entry\_covers\_log2size\_in\_bytes**

#### **Type**

uint32\_t

#### **Default value**

30

Each LOGPT entry covers:

```
2**rme_10gpt_entry_covers_log2size_in_bytes
```

bytes of address space.

The valid values for this parameter are: 30, 34, 36, 39

This parameter is reported in an encoded format as the read-only field:

```
SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ
```

This parameter can be overridden by the port `10gptsz_s` when sampled on negedge of reset.

### **rme\_speculation\_control**

#### **Type**

string

#### **Default value**

“”

This is a comma-separated list of flags that control when and how the model will perform speculation for RME.

**root\_register\_page\_offset****Type**

uint64\_t

**Default value**

0

This is the offset from SMMU\_BASE of the Root register file page which is 64 KiB in size. It must not overlap any other part of the register map.

**secure\_state\_controls\_access\_to\_SMMU\_S\_INIT****Type**

bool

**Default value**

true

With RME access control of the SMMU\_S\_INIT belongs to Root. This parameters allows Root to delegate access control to the secure state, enabling secure software to reset the TLB, clearing out any TLB entries.

If RME is implemented and this parameter is 0, allow\_non\_secure\_access\_to\_SMMU\_S\_INIT has no effect.

If the SMMU does not implement RME then this parameter is ignored.

**seed****Type**

uint32\_t

**Default value**

0x12345678

Used to seed the pseudo-random number generator that the SMMU model uses.

**separate\_tw\_msi\_qs\_port****Type**

bool

**Default value**

true

True if there is a separate port which is used to walk configuration tables, translation tables, issue MSIs and access the queues. If this is false then pvbus\_m[0] will be used.

**size\_of\_cd\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding CD structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_dpttlb****Type**

uint32\_t

**Default value**

0

The number of entries in the DPT TLB. If this is zero then it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_gpttlb****Type**

uint32\_t

**Default value**

0

The number of entries in the GPT TLB. If this is zero then it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_llcd\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding L1CD descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_llste\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding L1STE descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_register\_file****Type**

uint64\_t

**Default value**

0x100000

This is the power of two size that the register file occupies in the memory map. It is used to generate a mask for the addresses received on pvbus\_control\_s to decode the desired register offset.

The default for this parameter is 1 MiB.

**size\_of\_ste\_cache****Type**

uint32\_t

**Default value**

0

The number of entries in the cache holding STE structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**size\_of\_tlb****Type**

uint32\_t

**Default value**

0

The number of entries in the TLB. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

**smmu\_msi\_device\_id****Type**

uint32\_t

**Default value**

0

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID.

See parameters `msi_attribute_transform` and `enable_device_id_checks`.

**smmu33\_begin\_offset\_of\_qcp0****Type**

uint32\_t

**Default value**

-

This is the offset from SMMU\_BASE of the first QCP page. The architecture requires that if more than one world of QCPs are present then they are in the order non-secure and then secure QCPs and form one continuous address space in the register file.

**support\_for\_httu\_when\_starts\_disallowed****Type**

unsigned

**Default value**

0

SMMU\_IDR0.HTTU describes to the programmer whether the SMMU and system support HTTU. Typically an SMMU that is capable of HTTU will have a configuration pin that says whether the system supports HTTU or not.

The SMMU model determines SMMU\_IDR0.HTTU as follows: \* If the parameter SMMU\_IDR0 indicates any kind of support for HTTU, then the configuration pin turns support on and off between that value and no support for HTTU. \* If the parameter SMMU\_IDR0 indicates no HTTU support, allow the pin to turn on support to that specified by this parameter.

Values for this parameter are the same as for the SMMU\_IDR0.HTTU field: \* 0 – no support for HTTU \* 1 – AF flag only \* 2 – AF flag and DBM update

**tlb\_when\_do\_f\_tlb\_conflict\_on\_overlap****Type**

unsigned

**Default value**

0

If a TLB entry is created by a walk and it overlaps an existing entry, there are some architectural situations where the result is known. For all others, then an implementation is allowed to use an **UNPREDICTABLE** combination of the two entries, or it can generate F\_TLB\_CONFLICT:

**0**

never generate

**1**

sometimes generate

**2**

always generate

Conflicts between global and non-global entries are not detected by the model.

### **translated\_device\_id\_base**

#### **Type**

uint32\_t

#### **Default value**

0

When appropriately enabled, assume that client device accesses are translated to a DeviceID as seen by the GIC of:

```
StreamID + translated_device_id_base
```

See parameter `output_attribute_transform` and `enable_device_id_checks`.

### **treat\_debug\_read\_accesses\_as\_speculative\_accesses**

#### **Type**

bool

#### **Default value**

false

The SMMU architecture has the concept of speculative accesses. If you set this flag to true, then debug read accesses flowing from the upstream system through the SMMU will be interpreted as speculative.

The difference is that a speculative read will: \* participate in HTTU \* if it encounters a (non-HTTU) fault will always return abort

Debug writes are still considered as debug accesses. All speculative writes would be aborted and this is not a useful behaviour for the SMMU to emulate.

### **tw\_qs\_attribute\_transform**

#### **Type**

string

#### **Default value**

""

Transform downstream attributes of table walk and queue transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / MasterID / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**HWATTR\_KIND\_0**

PBHA information

**kind**

Transaction kind

- For a read:
  - 0/1**  
L1STE/STE
  - 2/3**  
L1CD/CD
  - 4/5**  
S1/S2 TTD (including CAS)
  - 6**  
CMDQ
  - 7**  
VMS
  - 11/12**  
LOGPT/L1GPT
  - 13/14**  
LODPT/L1DPT
- For a write:
  - 0**  
EVENTQ
  - 1**  
PRIQ

ExtendedID/ManagerID64/MasterID/UserFlags start with values {0, 0xFFFFFFFF, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.

**Note**

See also `output_attribute_transform` and `msi_attribute_transform`.

---

**`unpred_httu_percent_do_discretionary_AF`****Type**

unsigned

**Default value**

50

If a descriptor could have a discretionary update of the AF flag on then what is the percentage of the time that the AF update should occur.

**`unpred_httu_percent_do_discretionary_DBM`****Type**

unsigned

**Default value**

50

If a descriptor could have a discretionary DBM update to make the descriptor WriteableDirty then what is the percent of the time time that the DBM update should occur.

**`unpred_translated_access_out_of_range_of_oas`****Type**

unsigned

**Default value**

1

If a Translated Access is presented to the SMMU that is > OAS then it is CONSTRAINED UNPRED as to whether the transaction will either: 0 – be truncated to OAS and go downstream 1 – be aborted, no event written

**`wait_atos_ticks`****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing an ATOS operation. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time [0,max(2,(t & 0xFFFFffff))-1].



**wait\_cmdq\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing something on the command queue. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffff)) - 1]$ .

**wait\_eventq\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing something on the event queue. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffff)) - 1]$ .

**wait\_misc\_async\_actions\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before doing an async action that could be delayed is performed. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffff)) - 1]$ .

**wait\_msi\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before sending an MSI. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFffff)) - 1]$ .

**wait\_pri\_req\_ticks****Type**

uint64\_t

**Default value**

0

This is the time to wait before processing a PRI Request. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

**wait\_pri\_resp\_ticks****Type**

uint64\_t

**Default value**

1

This is the time to wait before sending a PRI Response back to the PCIe subsystem. When a PRI Response is an auto-response then the ATC might immediately make a new ATS request, that immediately fails, immediately makes a PRI Request, or auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

**when\_fetch\_vms****Type**

unsigned

**Default value**

0

Architecturally, there is flexibility in how a VMS is cached and thus: \* when it will be fetched \* the prioritization of F\_VMS\_FETCH.

Of the many architecturally-allowed options, the model offers two:

0 – fetched and cached immediately after the STE is fetched  
1 – fetched and cached immediately after the CD is fetched

In both cases, then the VMS is cached in the STE and CMD\_CFGI\_VMS\_PIDM is a **NOP**.

**width\_of\_agbpa\_impdef****Type**

uint32\_t

**Default value**

16

Width of the SMMU\_s\_AGBPA.IMPDEF field.

### 3.10.101 SMSC\_91C111

10/100 Non-PCI Ethernet Controller(SMSC 91C111). This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1249: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About SMSC\_91C111

This component provides the register interface of the SMSC part and can be configured to act as an unconnected Ethernet port, or an Ethernet port connected to the host by an Ethernet bridge.

It uses a banked register model of primarily 16-bit registers. There are also indirectly accessible registers for the PHY unit.

If a MAC address is not specified in the `mac_address` parameter, the simulator takes the default MAC address, which is randomly generated. This provides some degree of MAC address uniqueness when running models on multiple hosts on a local network.



DHCP servers allocate the IP addresses, but because they sometimes do this based on the MAC address provided to them, using random MAC addresses might interact unfortunately with some DHCP servers.

See also:

- [Configuring the networking environment for Linux](#)

#### Iris and MTI instances for SMSC\_91C111

This model has the following Iris instances:

**Table 3-1250: SMSC\_91C111 Iris instances**

InstanceName	ComponentName
SMSC_91C111	SMSC_91C111
SMSC_91C111.SMSC_slave	PVBusSlave

This model has the following MTI trace components:

**Table 3-1251: SMSC\_91C111 MTI instances**

InstanceName	ComponentName
SMSC_91C111.SMSC_slave	PVBusSlave

## Ports for SMSC\_91C111

Table 3-1252: Ports

Name	Protocol	Type	Description
clock	<a href="#">ClockSignal</a>	Slave	Clock input, typically 25MHz, which sets the master transmit/receive rate.
eth	<a href="#">VirtualEthernet</a>	Master	Ethernet port.
intr	<a href="#">Signal</a>	Master	Interrupt signal.
pvbus	<a href="#">PVBUS</a>	Slave	Slave port for register access.
state	<a href="#">ValueState_64</a>	Master	State port to retrieve state of host bridge

## Parameters for SMSC\_91C111

**cache\_size****Type**

int

**Default value**

0x10000

**Description**

Size of cache memory in SMSC MMU.

**enabled****Type**

bool

**Default value**

0x0

**Description**

Host interface connection enabled.

**mac\_address****Type**

string

**Default value**

"00:02:f7:ef:00:00"

**Description**

Host/model MAC address.

**not\_lan911x****Type**

bool

**Default value**

0x0

**Description**

Gracefully fail SMSC LAN911x driver probe.

**promiscuous****Type**

bool

**Default value**

0x1

**Description**

Put host into promiscuous mode.

### 3.10.102 SP804\_Timer

ARM Dual-Timer Module(SP804). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1253: IP revisions support**

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Iris and MTI instances for SP804\_Timer**

This model has the following Iris instances:

**Table 3-1254: SP804\_Timer Iris instances**

InstanceName	ComponentName
SP804_Timer	SP804_Timer
SP804_Timer.busslave	PVBusSlave
SP804_Timer.clk_div0	ClockDivider
SP804_Timer.clk_div1	ClockDivider
SP804_Timer.counter0	CounterModule
SP804_Timer.counter1	CounterModule

This model has the following MTI trace components:

**Table 3-1255: SP804\_Timer MTI instances**

InstanceName	ComponentName
SP804_Timer.busslave	PVBusSlave
SP804_Timer.clk_div0	ClockDivider
SP804_Timer.clk_div1	ClockDivider

## Ports for SP804\_Timer

Table 3-1256: Ports

Name	Protocol	Type	Description
clock	<a href="#">ClockSignal</a>	Slave	Clock input, typically 1MHz, driving master count rate.
irq_out0	<a href="#">Signal</a>	Master	Interrupt signaling.
irq_out1	<a href="#">Signal</a>	Master	Interrupt signaling.
pvbuss	<a href="#">PVBuss</a>	Slave	Slave port for register access.
timer_en[2]	<a href="#">ClockRateControl</a>	Slave	Port for changing the rate of timer n.

## Parameters for SP804\_Timer

**clk\_div0.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clk\_div0.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clk\_div1.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clk\_div1.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**3.10.103 SP805\_Watchdog**

ARM Watchdog Module(SP805). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1257: IP revisions support**

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Iris and MTI instances for SP805\_Watchdog**

This model has the following Iris instances:

**Table 3-1258: SP805\_Watchdog Iris instances**

InstanceName	ComponentName
SP805_Watchdog	SP805_Watchdog
SP805_Watchdog.busslave	PVBusSlave
SP805_Watchdog.clocktimer	ClockTimerThread
SP805_Watchdog.clocktimer.timer	ClockTimerThread64
SP805_Watchdog.clocktimer.timer.thread	SchedulerThread
SP805_Watchdog.clocktimer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

**Table 3-1259: SP805\_Watchdog MTI instances**

InstanceName	ComponentName
SP805_Watchdog.busslave	PVBusSlave

**Ports for SP805\_Watchdog****Table 3-1260: Ports**

Name	Protocol	Type	Description
clk_in	<a href="#">ClockSignal</a>	Slave	Clock input, typically 1MHz, driving master count rate.
irq_out	<a href="#">Signal</a>	Master	Interrupt signaling.
pvbus_s	<a href="#">PVBus</a>	Slave	Slave port for register access.
reset_in	<a href="#">Signal</a>	Slave	Reset signaling.
reset_out	<a href="#">Signal</a>	Master	Reset signaling.

## Parameters for SP805\_Watchdog

### **simhalt**

#### Type

bool

#### Default value

0x0

#### Description

Halt on reset.

## 3.10.104 SP810\_SysCtrl

PrimeXsys System Controller(SP810) NB: Only EB relevant functionalities are fully implemented.  
This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1261: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Iris and MTI instances for SP810\_SysCtrl

This model has the following Iris instances:

**Table 3-1262: SP810\_SysCtrl Iris instances**

InstanceName	ComponentName
SP810_SysCtrl	SP810_SysCtrl
SP810_SysCtrl.busslave	PVBusSlave
SP810_SysCtrl.clkdiv_clk0	ClockDivider
SP810_SysCtrl.clkdiv_clk1	ClockDivider
SP810_SysCtrl.clkdiv_clk2	ClockDivider
SP810_SysCtrl.clkdiv_clk3	ClockDivider

This model has the following MTI trace components:

**Table 3-1263: SP810\_SysCtrl MTI instances**

InstanceName	ComponentName
SP810_SysCtrl.busslave	PVBusSlave
SP810_SysCtrl.clkdiv_clk0	<a href="#">ClockDivider</a>
SP810_SysCtrl.clkdiv_clk1	<a href="#">ClockDivider</a>



InstanceName	ComponentName
SP810_SysCtrl.clkdiv_clk2	ClockDivider
SP810_SysCtrl.clkdiv_clk3	ClockDivider

## Ports for SP810\_SysCtrl

**Table 3-1264: Ports**

Name	Protocol	Type	Description
clk_in	ClockSignal	Slave	Clock input.
hclkdivsel	ValueState	Master	Define the processor clock/bus clock ratio. Not fully implemented. Using this port has unpredictable results.
npwr	Signal	Slave	Power on reset. Not fully implemented. Using this port has unpredictable results.
pll_en	Signal	Master	PLL enable output. Not fully implemented. Using this port has unpredictable results.
pvbuss	PVBus	Slave	Slave port for register access.
ref_clk_in	ClockSignal	Slave	Clock source used by the Timer and Watchdog modules.
remap_clear	StateSignal	Master	Remap clear request output.
remap_stat	StateSignal	Slave	Remap status input. Not fully implemented. Using this port has unpredictable results.
sleep_mode	Signal	Master	Control clocks for SLEEP mode. Not fully implemented. Using this port has unpredictable results.
sys_id	ValueState	Slave	Unused port.
sys_mode	ValueState	Slave	Present system mode. Not fully implemented. Using this port has unpredictable results.
sys_stat	ValueState	Slave	System status input. Not fully implemented. Using this port has unpredictable results.
timer_clk_en[4]	ClockRateControl	Master	Timer clock enable n.
wd_clk_en	Signal	Master	Watchdog module clock enable output. Not fully implemented. Using this port has unpredictable results.
wd_en	Signal	Slave	Watchdog module enable input. Not fully implemented. Using this port has unpredictable results.

## Parameters for SP810\_SysCtrl

### clkdiv\_clk0.div

#### Type

int

#### Default value

0x1

#### Description

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

### clkdiv\_clk0.mul

#### Type

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv\_clk1.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv\_clk1.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv\_clk2.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv\_clk2.mul****Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv\_clk3.div****Type**

int

**Default value**

0x1

**Description**

Clock Rate Divider. This parameter is not exposed via CADI and can only be set in LISA.

**clkdiv\_clk3.mul**

**Type**

int

**Default value**

0x1

**Description**

Clock Rate Multiplier. This parameter is not exposed via CADI and can only be set in LISA.

**sysid**

**Type**

int

**Default value**

0x0

**Description**

System Identification Register.

**use\_s8**

**Type**

bool

**Default value**

0x0

**Description**

Use Switch 8 (S1-S4).

**3.10.105 SSU**

Safety Status Unit. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1265: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Iris and MTI instances for SSU

This model has the following Iris instances:

**Table 3-1266: SSU Iris instances**

InstanceName	ComponentName
SSU	SSU
SSU.pvbus_slave	PVBusSlave

This model has the following MTI trace components:

**Table 3-1267: SSU MTI instances**

InstanceName	ComponentName
SSU	SSU
SSU.pvbus_slave	PVBusSlave

## Ports for SSU

**Table 3-1268: Ports**

Name	Protocol	Type	Description
c_error_in	Signal	Slave	Critical interrupts to the SSU
clk_in	ClockSignal	Slave	Clock input
cold_reset_in	Signal	Slave	Cold reset signal
nc_error_in	Signal	Slave	Non critical interrupts to the SSU
pvbus_s	PVBus	Slave	To access FMU model registers
ssu_out	ValueState	Master	SSU output port
warm_reset_in	Signal	Slave	Warm reset signal

## Parameters for SSU

### diagnostics

#### Type

int

#### Default value

0x2

#### Description

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT:2).

### sm\_implemented

#### Type

bool

#### Default value

0x0

**Description**

Safety Mechanism Implemented.

**3.10.106 SecureAlarmManager**

Security Alarm Manager. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1269: IP revisions support**

Revision	Quality level
0.93	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

**Iris and MTI instances for SecureAlarmManager**

This model has the following Iris instances:

**Table 3-1270: SecureAlarmManager Iris instances**

InstanceName	ComponentName
SecureAlarmManager	SecureAlarmManager
SecureAlarmManager.apb	PVBusSlave

This model has the following MTI trace components:

**Table 3-1271: SecureAlarmManager MTI instances**

InstanceName	ComponentName
SecureAlarmManager.apb	PVBusSlave

**Ports for SecureAlarmManager****Table 3-1272: Ports**

Name	Protocol	Type	Description
apb	PVBus	Slave	APB Subordinate Interface - Access to registers
clk_in	ClockSignal	Slave	Clock in signal
config_done_trig_ack_in	Signal	Slave	Config done ack signal
config_done_trig_req_out	Signal	Master	Config done req signal
event_in[61]	Signal	Slave	Event in signal
event_status_out[64]	Signal	Master	Event status out signal
nCOLDRESETAON_in	Signal	Slave	Coldreset in signal
reset_in	Signal	Slave	Reset in signal
response_action_out[8]	Signal	Master	Response action out signal

Parameters for SecureAlarmManager

NUM\_SAMNEC

Type  
int

Default value  
0x3

Description  
Number of SAM event counters.

NUM\_SAMNRA

Type  
int

Default value  
0x7

Description  
Number of SAM response actions.

diagnostics

Type  
int

Default value  
0x2

Description  
Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

3.10.107 SecureCache

SecureCache. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

Table 3-1273: IP revisions support

Revision	Quality level
1.13	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

Iris and MTI instances for SecureCache

This model has the following Iris instances:

**Table 3-1274: SecureICache Iris instances**

InstanceName	ComponentName
SecureICache	SecureICache
SecureICache.DECRYPT_RAM_0	PVBusSlave
SecureICache.DECRYPT_RAM_1	PVBusSlave
SecureICache.HTR_RAM	PVBusSlave
SecureICache.SIC_BusMapper	PVBusMapper
SecureICache.SIC_BusMapper0	PVBusMapper
SecureICache.SIC_BusMapper1	PVBusMapper
SecureICache.ZERO_RAM	PVBusSlave
SecureICache.apb	PVBusSlave

This model has the following MTI trace components:

**Table 3-1275: SecureICache MTI instances**

InstanceName	ComponentName
SecureICache.DECRYPT_RAM_0	PVBusSlave
SecureICache.DECRYPT_RAM_1	PVBusSlave
SecureICache.HTR_RAM	PVBusSlave
SecureICache.SIC_BusMapper	PVBusMapper
SecureICache.SIC_BusMapper0	PVBusMapper
SecureICache.SIC_BusMapper1	PVBusMapper
SecureICache.ZERO_RAM	PVBusSlave
SecureICache.apb	PVBusSlave

## Ports for SecureICache

**Table 3-1276: Ports**

Name	Protocol	Type	Description
apb	PVBus	Slave	APB4 Subordinate Interface - Access to registers
htr_ram_pvbus_s	PVBus	Slave	To Read and Write pre-calculated SHA-256 Digests
irq_out	Signal	Master	To indicate interrupts
pvbus_m	PVBus	Master	To read from External RAM
pvbus_s	PVBus	Slave	Receives Transactions from CPU
reset_in	Signal	Slave	reset

## Parameters for SecureICache

### SIC\_AUTH\_ENABLE

#### Type

bool

**Default value**

0x0

**Description**

SIC Authentication enabled [default==false].

**SIC\_DECRYPT\_ENABLE****Type**

bool

**Default value**

0x0

**Description**

SIC Decryption enabled [default==false].

**SIC\_DR\_CNT****Type**

int

**Default value**

0x1

**Description**

SIC config: Decryption Region Count (0x0==1, 0x1==2[default], 0x2==4).

**SIC\_HTR\_RAM\_SIZE****Type**

int

**Default value**

0x20

**Description**

SIC config: Hash Tag RAM Size (0x1==1KB, 0x2==2KB, 0x4==4KB, 0x8==8KB, 0x10=16KB, 0x20=32KB[default]).

**SIC\_MAX\_CODE\_SIZE****Type**

int

**Default value**

0x400

**Description**

SIC config: Maximum Code Size supported (n==nKB[default,1024(1MB)]).



**SIC\_PAGE\_RAM\_SIZE****Type**

int

**Default value**

0x10

**Description**

SIC config: Page RAM Size (0x4==4KB, 0x8==8KB[default], 0x10=16KB).

**SIC\_PAGE\_SIZE****Type**

int

**Default value**

0x3

**Description**

SIC config: Page Size (0x0==128B, 0x1==256B, 0x2=512B, 0x3=1KB[default], 0x4=2KB, 0x5=4KB).

**SIC\_PMON\_EN****Type**

bool

**Default value**

0x0

**Description**

SIC config: Performance Montior enable [default==false].

**diagnostics****Type**

int

**Default value**

0x2

**Description**

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

### 3.10.108 SystemFMU

System level Fault Management Unit. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1277: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Iris and MTI instances for SystemFMU

This model has the following Iris instances:

**Table 3-1278: SystemFMU Iris instances**

InstanceName	ComponentName
SystemFMU	SYSTEM_FMU
SystemFMU.pvbus_slave	PVBusSlave

This model has the following MTI trace components:

**Table 3-1279: SystemFMU MTI instances**

InstanceName	ComponentName
SystemFMU	SYSTEM_FMU
SystemFMU.pvbus_slave	PVBusSlave

## Ports for SystemFMU

**Table 3-1280: Ports**

Name	Protocol	Type	Description
c_error_in[26]	Signal	Slave	Critical interrupts from the device side FMUs
c_error_out	Signal	Master	Critical error input from the system FMU
clk_in	ClockSignal	Slave	Clock input
cold_reset_in	Signal	Slave	Cold reset signal
nc_error_in[26]	Signal	Slave	Non critical interrupts from the device side FMUs
nc_error_out	Signal	Master	Non critical error input from the system FMU
pvbus_s	PVBus	Slave	To access FMU model registers
warm_reset_in	Signal	Slave	Warm reset signal

## Parameters for SystemFMU

### diagnostics

#### Type

int

#### Default value

0x2

#### Description

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

**upstream\_fmus\_cfg****Type**

int

**Default value**

0x1

**Description**

Number of upstream FMUs that can be connected to the System level FMU.

### 3.10.109 System\_RAS\_Agent

System level RAS agent. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1281: IP revisions support**

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### Iris and MTI instances for System\_RAS\_Agent

This model has the following Iris instances:

**Table 3-1282: System\_RAS\_Agent Iris instances**

InstanceName	ComponentName
System_RAS_Agent	SYSTEM_RAS_AGENT
System_RAS_Agent.pvbus_subordinate	PVBusSlave

This model has the following MTI trace components:

**Table 3-1283: System\_RAS\_Agent MTI instances**

InstanceName	ComponentName
System_RAS_Agent.pvbus_subordinate	PVBusSlave

#### Ports for System\_RAS\_Agent

**Table 3-1284: Ports**

Name	Protocol	Type	Description
apb	PVBus	Slave	To access RAS agent registers
clk_in	ClockSignal	Slave	Input Clock - RAS agent is in this clock domain
cri_in[56]	Signal	Slave	Critical Error Interrupt from the downstream RAS agent
cri_out	Signal	Master	CRI_OUT is the consolidated status of the Critical Error Interrupt from this and the downstream RAS agent(s).

Name	Protocol	Type	Description
eri_in[56]	Signal	Slave	Error Recovery Interrupt from the downstream RAS agent
eri_out	Signal	Master	consolidated status of the Error Recovery Interrupt from this and the downstream RAS agent(s).
fhi_in[56]	Signal	Slave	Fault Handling Interrupt from the downstream RAS agent
fhi_out	Signal	Master	consolidated status of the Fault Handling Interrupt from this and the downstream RAS agent(s).
reset_in	Signal	Slave	Input reset - Connect to the system cold reset.
valid_in[56]	Signal	Slave	Incoming Valid from a downstream RAS agent indicates the presence of at least one valid error record.
valid_out	Signal	Master	outgoing Valid from the downstream RAS agent(s) contains at least one valid error record.

## Parameters for System\_RAS\_Agent

### NUM\_DOWNSTREAM\_RAS\_AGENTS

#### Type

int

#### Default value

0x1

#### Description

Number of downstream RAS agents for which the proxy error record is maintained.

### SYNC\_ENABLE

#### Type

int

#### Default value

0x0

#### Description

Enables synchronization on the Interrupt lines and valid line before assigning it to the corresponding bits in the ERR<n>STATUS register. 1-bit per downstream RAS Agent.

### diagnostics

#### Type

int

#### Default value

0x2

#### Description

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

### 3.10.110 TZC\_400

TrustZone Address Space Controller. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1285: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

#### About TZC\_400

The TZC-400 determines, under software control, whether a particular bus master is permitted to issue Non-secure accesses to a particular physical address.

The component has:

- Eight address regions in addition to the base region, region 0.
- A programmable control block for security-access permissions configuration through the Advanced Peripheral Bus (APB).
- Up to four address filters that share common set region set-up registers.
- Software configurable permission check failure reporting and interrupt signaling.
- Filtering with a Non-Secure Access ID (NSAID).
- A gate keeper, to allow or block accesses to the filter unit.
- Configurable reset values of region configuration registers and other key configuration registers.

This component has the following subcomponents:

#### TZFilterUnits

The TZC-400 has four TZFilterUnits. The `BUILD_CONFIG` register sets the configuration. The `rst_build_config` parameter controls the register. The value of `rst_build_config` varies with the system. See the system design documentation or system integration documentation. For AEMvA, it is `0x3003F08`.

#### TZDummyDevice

An internal dummy device that mimics **RAZ/WI** for TZFilterUnits. The system uses it when there is a permission violation and a bus returns Transaction OK.

#### Differences between the model and the RTL

Unlike the hardware, this component does not have:

- Asynchronous clocks. The model does not need clocks for data transfer, or clock signals.
- QoS Virtual Network (QVN) support. Specifically, it does not implement the `vnet` bits[27:24] in `FAIL_ID_<x>` registers.
- Fast Path and Fast Path ID. In the model, transactions occur at similar speeds.

- 256 outstanding accesses globally for each read or write Normal Paths and configurable 8, 16, or 32 outstanding accesses on Fast Path read access. The model does not support QVN. This concept is meaningless for a PV level model.
- Configurable address bus width, data bus width, transaction ID tag, and USER bus width. A single bus implementation, PVBUS, covers these AXI bus hardware implementation details.

This component does not implement:

- The vnet bits[27:24] in `FAIL_ID_<x>` registers.
- Any background logic for the speculation control register. This does not affect model behavior.

## Configuration

- Configure `master_id_from_label` OR `id_mapping`, `rst_build_config`, and `rst_region_attributes_0` before running the model to set the desired behaviors. Otherwise, the system resets all region configuration registers, `rst_action`, and `rst_gate_keeper` to 0, and resets `rst_build_config` and `rst_region_attributes_0` to sensible default values.
- Configure either `id_mapping` OR `master_id_from_label` at model init, or a warning message appears.
- The syntax of `id_mapping` is:

```
<masterid_0>:<nsaid_0>,<masterid_1>:<nsaid_1>,<masterid_n>:<nsaid_n>
```

Separate the mapping pairs by a comma. The masterid is the ID of the bus master, such as the parameter `CLUSTER_ID` on Cortex-A15/7, `cluster_id` port of Cortex-A15/7, or `master_id` parameter for Cortex-M3.

## Iris and MTI instances for TZC\_400

This model has the following Iris instances:

**Table 3-1286: TZC\_400 Iris instances**

InstanceName	ComponentName
TZC_400	TZC_400
TZC_400.apbslave[0]	PVBusSlave
TZC_400.filter0	filter0
TZC_400.filter0.BusMapper	PVBusMapper
TZC_400.filter1	filter1
TZC_400.filter1.BusMapper	PVBusMapper
TZC_400.filter2	filter2
TZC_400.filter2.BusMapper	PVBusMapper
TZC_400.filter3	filter3
TZC_400.filter3.BusMapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-1287: TZC\_400 MTI instances**

InstanceName	ComponentName
TZC_400.apbslave[0]	PVBusSlave
TZC_400.filter0.BusMapper	PVBusMapper
TZC_400.filter1.BusMapper	PVBusMapper
TZC_400.filter2.BusMapper	PVBusMapper
TZC_400.filter3.BusMapper	PVBusMapper

## Ports for TZC\_400

**Table 3-1288: Ports**

Name	Protocol	Type	Description
apbslave_s	PVBus	Slave	Bus access for control register.
filter_pvbus_m[4]	PVBus	Master	Outgoing bus traffic from filter units.
filter_pvbus_s[4]	PVBus	Slave	Incoming bus traffic to filter units.
tzc_reset	Signal	Slave	Reset signal from external master.
tzcint	Signal	Master	TrustZone interrupt signal, controlled by ACTION register.

## Parameters for TZC\_400

### diagnostics

#### Type

int

#### Default value

0x0

#### Description

Diagnostics.

### id\_mapping

#### Type

string

#### Default value

"0:0,1:0,2:0,3:0,4:0,5:0,6:0,7:0,8:0,9:0,10:0,11:0,12:0,13:0,14:0,15:0"

#### Description

Mapping from Master ID to NSAID.

### master\_id\_from\_label

#### Type

bool

#### Default value

0x0

**Description**

Obtain Master ID from label (ignores id\_mapping).

**rst\_action****Type**

int

**Default value**

0x0

**Description**

ACTION register value at reset.

**rst\_build\_config****Type**

int

**Default value**

0x3003f08

**Description**

BUILD\_CONFIG register value at reset.

**rst\_gate\_keeper****Type**

int

**Default value**

0x0

**Description**

GATE\_KEEPER register value at reset.

**rst\_region\_attributes\_0****Type**

int

**Default value**

0xf

**Description**

Region 0 Secure attributes.

**rst\_region\_attributes\_1****Type**

int

**Default value**

0x0



**Description**

Region 1 Secure attributes.

**`rst_region_attributes_2`****Type**

int

**Default value**

0x0

**Description**

Region 2 Secure attributes.

**`rst_region_attributes_3`****Type**

int

**Default value**

0x0

**Description**

Region 3 Secure attributes.

**`rst_region_attributes_4`****Type**

int

**Default value**

0x0

**Description**

Region 4 Secure attributes.

**`rst_region_attributes_5`****Type**

int

**Default value**

0x0

**Description**

Region 5 Secure attributes.

**`rst_region_attributes_6`****Type**

int

**Default value**

0x0

**Description**

Region 6 Secure attributes.

**`rst_region_attributes_7`****Type**

int

**Default value**

0x0

**Description**

Region 7 Secure attributes.

**`rst_region_attributes_8`****Type**

int

**Default value**

0x0

**Description**

Region 8 Secure attributes.

**`rst_region_base_high_1`****Type**

int

**Default value**

0x0

**Description**

Region 1 base memory address (high 32 bits).

**`rst_region_base_high_2`****Type**

int

**Default value**

0x0

**Description**

Region 2 base memory address (high 32 bits).

**`rst_region_base_high_3`****Type**

int

**Default value**

0x0

**Description**

Region 3 base memory address (high 32 bits).

**`rst_region_base_high_4`**

**Type**

int

**Default value**

0x0

**Description**

Region 4 base memory address (high 32 bits).

**`rst_region_base_high_5`**

**Type**

int

**Default value**

0x0

**Description**

Region 5 base memory address (high 32 bits).

**`rst_region_base_high_6`**

**Type**

int

**Default value**

0x0

**Description**

Region 6 base memory address (high 32 bits).

**`rst_region_base_high_7`**

**Type**

int

**Default value**

0x0

**Description**

Region 7 base memory address (high 32 bits).

**`rst_region_base_high_8`**

**Type**

int

**Default value**

0x0

**Description**

Region 8 base memory address (high 32 bits).

**rst\_region\_base\_low\_1**

**Type**

int

**Default value**

0x0

**Description**

Region 1 base memory address (low 32 bits).

**rst\_region\_base\_low\_2**

**Type**

int

**Default value**

0x0

**Description**

Region 2 base memory address (low 32 bits).

**rst\_region\_base\_low\_3**

**Type**

int

**Default value**

0x0

**Description**

Region 3 base memory address (low 32 bits).

**rst\_region\_base\_low\_4**

**Type**

int

**Default value**

0x0

**Description**

Region 4 base memory address (low 32 bits).

**rst\_region\_base\_low\_5**

**Type**

int

**Default value**

0x0

**Description**

Region 5 base memory address (low 32 bits).

**rst\_region\_base\_low\_6**

**Type**

int

**Default value**

0x0

**Description**

Region 6 base memory address (low 32 bits).

**rst\_region\_base\_low\_7**

**Type**

int

**Default value**

0x0

**Description**

Region 7 base memory address (low 32 bits).

**rst\_region\_base\_low\_8**

**Type**

int

**Default value**

0x0

**Description**

Region 8 base memory address (low 32 bits).

**rst\_region\_id\_access\_0**

**Type**

int

**Default value**

0x0

**Description**

Region 0 NSAID permissions.

**rst\_region\_id\_access\_1**

**Type**

int

**Default value**

0x0

**Description**

Region 1 NSAID permissions.

**rst\_region\_id\_access\_2**

**Type**

int

**Default value**

0x0

**Description**

Region 2 NSAID permissions.

**rst\_region\_id\_access\_3**

**Type**

int

**Default value**

0x0

**Description**

Region 3 NSAID permissions.

**rst\_region\_id\_access\_4**

**Type**

int

**Default value**

0x0

**Description**

Region 4 NSAID permissions.

**rst\_region\_id\_access\_5**

**Type**

int

**Default value**

0x0

**Description**

Region 5 NSAID permissions.

**rst\_region\_id\_access\_6**

**Type**

int

**Default value**

0x0

**Description**

Region 6 NSAID permissions.

**`rst_region_id_access_7`**

**Type**

int

**Default value**

0x0

**Description**

Region 7 NSAID permissions.

**`rst_region_id_access_8`**

**Type**

int

**Default value**

0x0

**Description**

Region 8 NSAID permissions.

**`rst_region_top_high_0`**

**Type**

int

**Default value**

0x0

**Description**

Region 0 (default) top memory address.

**`rst_region_top_high_1`**

**Type**

int

**Default value**

0x0

**Description**

Region 1 top memory address (high 32 bits).

**`rst_region_top_high_2`**

**Type**

int

**Default value**

0x0

**Description**

Region 2 top memory address (high 32 bits).

**`rst_region_top_high_3`**

**Type**

int

**Default value**

0x0

**Description**

Region 3 top memory address (high 32 bits).

**`rst_region_top_high_4`**

**Type**

int

**Default value**

0x0

**Description**

Region 4 top memory address (high 32 bits).

**`rst_region_top_high_5`**

**Type**

int

**Default value**

0x0

**Description**

Region 5 top memory address (high 32 bits).

**`rst_region_top_high_6`**

**Type**

int

**Default value**

0x0

**Description**

Region 6 top memory address (high 32 bits).

**`rst_region_top_high_7`**

**Type**

int

**Default value**

0x0



**Description**

Region 7 top memory address (high 32 bits).

**`rst_region_top_high_8`**

**Type**

int

**Default value**

0x0

**Description**

Region 8 top memory address (high 32 bits).

**`rst_region_top_low_1`**

**Type**

int

**Default value**

0x0

**Description**

Region 1 top memory address (low 32 bits).

**`rst_region_top_low_2`**

**Type**

int

**Default value**

0x0

**Description**

Region 2 top memory address (low 32 bits).

**`rst_region_top_low_3`**

**Type**

int

**Default value**

0x0

**Description**

Region 3 top memory address (low 32 bits).

**`rst_region_top_low_4`**

**Type**

int

**Default value**

0x0

**Description**

Region 4 top memory address (low 32 bits).

**`rst_region_top_low_5`**

**Type**

int

**Default value**

0x0

**Description**

Region 5 top memory address (low 32 bits).

**`rst_region_top_low_6`**

**Type**

int

**Default value**

0x0

**Description**

Region 6 top memory address (low 32 bits).

**`rst_region_top_low_7`**

**Type**

int

**Default value**

0x0

**Description**

Region 7 top memory address (low 32 bits).

**`rst_region_top_low_8`**

**Type**

int

**Default value**

0x0

**Description**

Region 8 top memory address (low 32 bits).

### 3.10.111 TZFilterUnit

TrustZone Filter Unit. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1289: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Iris and MTI instances for TZFilterUnit

This model has the following Iris instances:

**Table 3-1290: TZFilterUnit Iris instances**

InstanceName	ComponentName
TZFilterUnit	TZFilterUnit
TZFilterUnit.BusMapper	PVBusMapper

This model has the following MTI trace components:

**Table 3-1291: TZFilterUnit MTI instances**

InstanceName	ComponentName
TZFilterUnit.BusMapper	PVBusMapper

### Ports for TZFilterUnit

**Table 3-1292: Ports**

Name	Protocol	Type	Description
control	<a href="#">TZFilterControl</a>	Master	Configuration port.
pvbus_m	<a href="#">PVBus</a>	Master	Master bus port.
pvbus_s	<a href="#">PVBus</a>	Slave	Slave bus port.

## 3.10.112 TZIC

ARM TrustZone Interrupt Controller(SP890). This model is written in LISA+.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1293: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### About TZIC

The TZIC provides a software interface to the secure interrupt system in a TrustZone design. It provides secure control of the nFIQ and masks out the interrupt sources chosen for nFIQ from the interrupts that are passed onto a non-secure interrupt controller.

## Iris and MTI instances for TZIC

This model has the following Iris instances:

**Table 3-1294: TZIC Iris instances**

InstanceName	ComponentName
TZIC	TZIC
TZIC.busslave	PVBusSlave

This model has the following MTI trace components:

**Table 3-1295: TZIC MTI instances**

InstanceName	ComponentName
TZIC.busslave	PVBusSlave

## Ports for TZIC

**Table 3-1296: Ports**

Name	Protocol	Type	Description
fiq_out	Signal	Master	FIQ interrupt to processor.
input[32]	Signal	Slave	32 interrupt input sources.
irq_out[32]	Signal	Master	32 IRQ output ports.
nsfiq_in	Signal	Slave	Connects to the nFIQ output of the non-secure interrupt controller.
pvbus	PVBus	Slave	Slave port for connection to PV bus master/decoder.
sfiq_in	Signal	Slave	Daisy chaining secure FIQ input, otherwise connects to logic 1 if interrupt controller not daisy chained.

## 3.10.113 TrustedRAM

Trusted RAM. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1297: IP revisions support**

Revision	Quality level
0.4	Alpha support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Iris and MTI instances for TrustedRAM

This model has the following Iris instances:

**Table 3-1298: TrustedRAM Iris instances**

InstanceName	ComponentName
TrustedRAM	TrustedRAM

InstanceName	ComponentName
TrustedRAM.apb	PVBusSlave

This model has the following MTI trace components:

**Table 3-1299: TrustedRAM MTI instances**

InstanceName	ComponentName
TrustedRAM.apb	PVBusSlave

## Ports for TrustedRAM

**Table 3-1300: Ports**

Name	Protocol	Type	Description
apb	PVBus	Slave	APB Subordinate Interface - Access to registers
reset_in	Signal	Slave	Reset in signal

## Parameters for TrustedRAM

### TRBC\_RESET\_VALUE

#### Type

int

#### Default value

0xb

#### Description

TRBC Registers Reset Value.

### diagnostics

#### Type

int

#### Default value

0x2

#### Description

Diagnostics 0-4 (0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG - DEFAULT==2).

## 3.10.114 VHT\_VIOBridge

Arm VHT Virtual I/O interface. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1301: IP revisions support**

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Ports for VHT\_VIOBridge

**Table 3-1302: Ports**

Name	Protocol	Type	Description
pvbuss_s	PVBus	Slave	Target port for access to VIOBridge registers

### Parameters for VHT\_VIOBridge

#### **vio\_basename**

##### Type

string

##### Default value

""

##### Description

Basename of scripts to use for VIO.

#### **vio\_path**

##### Type

string

##### Default value

""

##### Description

Path to find python scripts for VIO.

## 3.10.115 VHT\_VSIBridge

Arm VHT Virtual stream interface. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1303: IP revisions support**

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

## Ports for VHT\_VSIBridge

**Table 3-1304: Ports**

Name	Protocol	Type	Description
intr	Signal	Master	Interrupt raising signal
pdbus_m	PVBus	Master	Requester port for access to ememory
pdbus_s	PVBus	Slave	Completer port for access to VSIBridge registers

## Parameters for VHT\_VSIBridge

### **vsi\_basename**

#### Type

string

#### Default value

""

#### Description

Basename of scripts to use for VSI.

### **vsi\_idx**

#### Type

int

#### Default value

0x0

#### Description

Index to use for VSI.

### **vsi\_path**

#### Type

string

#### Default value

""

#### Description

Path to find python scripts for VSI.

## 3.10.116 VHT\_VSocket

Arm VHT Virtual socket bridge interface. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1305: IP revisions support**

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Ports for VHT\_VSocket

**Table 3-1306: Ports**

Name	Protocol	Type	Description
pvbuss_m	PVBus	Master	Port for VSocket bridge to access external memory
pvbuss_s	PVBus	Slave	Target port for access to VSocket bridge registers

### Parameters for VHT\_VSocket

#### name

#### Type

string

#### Default value

""

#### Description

Name of the component.

## 3.10.117 v7\_VGIC

System VGIC architecture version v7. This model is written in C++.

This model supports the following revisions of the IP at the given quality levels:

**Table 3-1307: IP revisions support**

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [1.3 Quality level definitions](#) on page 22.

### Iris and MTI instances for v7\_VGIC

This model has the following Iris instances:

**Table 3-1308: v7\_VGIC Iris instances**

InstanceName	ComponentName
v7_VGIC	v7_VGIC
v7_VGIC.vgic_bus_slave	PVBusSlave

This model has the following MTI trace components:



**Table 3-1309: v7\_VGIC MTI instances**

InstanceName	ComponentName
v7_VGIC	v7_VGIC
v7_VGIC.vgic_bus_slave	PVBusSlave

## Ports for v7\_VGIC

**Table 3-1310: Ports**

Name	Protocol	Type	Description
cfgsdisable	Signal	Slave	Disable write access to some GIC registers.
configuration	v7_VGIC_Configuration_Protocol	Slave	Configure the mapping of the core number (from MasterID) to the core interface number.
fiq_in[8]	Signal	Slave	FIQ inputs.
fiq_out[8]	Signal	Master	FIQ outputs.
irq_in[8]	Signal	Slave	IRQ inputs.
irq_out[8]	Signal	Master	IRQ outputs.
ppi_core0[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_core1[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_core2[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_core3[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_core4[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_core5[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_core6[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_core7[16]	Signal	Slave	Private peripheral interrupts (ID16-ID31) for cpu 7.
pvbus_s	PVBus	Slave	Bus port for accessing distributor registers.
reporting_interface	VGICReportingProtocol	Slave	Logging interface.
reset_signal	Signal	Slave	Reset signal input.
spi[988]	Signal	Slave	SPI inputs.
vfiq_out[8]	Signal	Master	Virtual FIQ outputs.
virq_out[8]	Signal	Master	Virtual IRQ outputs.
wakeup_fiq[8]	Signal	Master	Wakeup signal for FIQ.
wakeup_irq[8]	Signal	Master	Wakeup signal for IRQ.

## Parameters for v7\_VGIC

### core-impl-id

#### Type

int

#### Default value

0x3902043b

#### Description

Implementation ID to present for the cores.

**dist-impl-id****Type**

int

**Default value**

0x3902043b

**Description**

Implementation ID to present for the distributor.

**enable\_log\_errors****Type**

bool

**Default value**

0x0

**Description****enable\_log\_fatal****Type**

bool

**Default value**

0x0

**Description****enable\_log\_warnings****Type**

bool

**Default value**

0x0

**Description****enabled****Type**

bool

**Default value**

0x1

**Description**

Enable the component. If it is disabled then all register writes will have no effect.

**number-of-cores****Type**

int

**Default value**

0x8

**Description**

Number of core interfaces to present.

**number-of-ints****Type**

int

**Default value**

0xe0

**Description**

Number of interrupt pins. Will be rounded down to the nearest multiple of 32.

**number-of-lrs****Type**

int

**Default value**

0x40

**Description**

Number of list registers.

**vgic-version****Type**

int

**Default value**

0xb

**Description**

Version number of the VGIC interface.

## 4. Plug-ins for Fast Models

This chapter describes the plug-ins that are available for Fast Models.

Prebuilt plug-ins can be found at `$PVLIB_HOME/plugins/<OS_compiler>/`. The source code for some of these plug-ins is provided as programming examples, under `$PVLIB_HOME/examples/MTI/`.

### 4.1 Loading a plug-in

The method of loading a plug-in depends on the type of model being used.

#### 4.1.1 --plugin command-line option

To load a plug-in, use the `--plugin <path_to_plugin>/<plugin_name>` option.

To load more than one plug-in, use multiple `--plugin` options.

Specify plug-in parameters using one or more `-c` options when launching the model, or use a configuration file.

For example the following command:

- Loads the TarmacTrace plug-in.
- Sets the TarmacTrace `trace-file` parameter.

```
./isim_system --plugin $plugin_dir/TarmacTrace.so -C TRACE.TarmacTrace.trace-  
file=tTrace.log
```

#### 4.1.2 scx::scx\_load\_plugin() method

Fast Models that are exported to SystemC can call the method `scx::scx_load_plugin()` to hard code the path to the plug-in, before calling `sc_start()`.

For example:

```
scx::scx_load_plugin("$PVLIB_HOME/plugins/Linux64_GCC-10.3/TarmacTrace.so");
```

#### Related information

- [scx::scx\\_load\\_plugin](#)

### 4.1.3 FM\_TRACE\_PLUGINS environment variable

If it is not possible to specify a trace plug-in to the launching tool, use the environment variable `FM_TRACE_PLUGINS`. This variable must be set to the full path of the plug-in.

For example, on Linux:

```
export FM_TRACE_PLUGINS=<installation_path>/plugins/<OS_Compiler>/TarmacTrace.so
```

or on Windows:

```
set FM_TRACE_PLUGINS=<installation_path>\plugins\<OS_Compiler>\<version>\TarmacTrace.dll
```

To set multiple trace plug-ins at the same time, separate them with semicolons, for example:

```
set FM_TRACE_PLUGINS=%PVLIB_HOME%\plugins\Win64_VC2019\Release\TarmacTrace.dll;
%PVLIB_HOME%\plugins\Win64_VC2019\Release\GenericTrace.dll
```

You can also load the same plug-in multiple times. Give each instance a name by adding a prefix `<instancename>=` to each plug-in path or paths.

Specify parameters for plug-ins that you load using `FM_TRACE_PLUGINS`, using the following syntax:

```
|<param0_without_prefix=value>||<param1_without_prefix=value>||
<paramN_without_prefix=value>|<absolute_path_to_plugin.dll>
```

where `<param*_without_prefix=value>` means you must specify the parameter without the prefix that would be used on the command line. For example, use `trace-file=file.txt` instead of `TRACE.TarmacTrace.trace-file=file.txt`.

For example, on Linux:

```
export FM_TRACE_PLUGINS='|trace-file=/home/work/trace.txt||end-instruction-
count=1000|$PVLIB_HOME/plugins/Linux64_GCC-9.3/TarmacTrace.so'
```

or on Windows:

```
set FM_TRACE_PLUGINS=^|trace-file=c:\work\trace.txt^|^|end-instruction-count=1000^|
%PVLIB_HOME%\plugins\Win64_VC2019\Release\TarmacTrace.dll
```



Note

- Do not specify a pipe character `|` at the end of the environment variable value.
- Separate individual parameters with two pipe characters `||`, not one.
- Separate the plug-in from the parameters using a single pipe character `|`.
- Do not specify quotes around paths that contain spaces, or escape spaces. Spaces are resolved automatically.

- Do not specify environment variables within this environment variable.
- Specify the parameters before you specify the plug-in.
- On Windows, you can either set the environment variable using the **Advanced System Settings** window, or if you are using the command line, you might need to escape each pipe character, for example `^|`.
- The parameter values that you set using this environment variable are not displayed by the `--list-params` or `-l` command-line options. This is because the environment variable is processed at a later stage, when launching the simulation.

---

## 4.2 Customizing a plug-in

You can customize the behavior of a plug-in using parameters.

Plug-in parameters are set in the same way as model parameters. They have the following format:

```
-C <PLUGIN_TYPE>.<plugin_name>.<parameter>=<value>
```

The `PLUGIN_TYPE` value varies depending on the plug-in type, for example:

```
-C TRACE.TarmacTrace.trace-file=trace.log
```

```
-C CRYPTO.Crypto.verbose=1
```

## 4.3 ArchMsgTrace

The Architecture Message Trace plug-in prints warning and error messages to `stdout` or to a file when software performs operations that are not recommended, for instance because they are unpredictable.

The plug-in connects to all trace sources that have the `ArchMsg` prefix, which are normal MTI trace sources, but with a specific format. These trace sources can also be used with the `GenericTrace` plug-in, but the `ArchMsgTrace` plug-in has extra capabilities.

When the model emits an `ArchMsg` trace event, `ArchMsgTrace` outputs a message in the format:

```
category: component.hierarchy.name: ...
```

The trace sources have names of the form:

```
component.hierarchy.ArchMsg.category.name[#supplementalEventName]
```

where:

- `category` is usually `Warning`, `Error`, or `Info`.
- `name` is a short string that uniquely identifies the condition.
- `supplementalEventName` is an optional identifier for a supplemental event, which is an event that provides more information about the initial event. For example, if a cache contains mismatching attributes, which triggers an `ArchMsg` trace event, a supplemental event might be emitted for each cache line affected.

The trace source can also include a line that defines a more human-readable description of the event. This line can contain fields which `ArchMsgTrace` replaces in the output string with values from the trace source.

`ArchMsgTrace` can be configured to suppress:

- Specific trace sources.
- Specific categories.
- Repeated events of the same type.

To suppress specific trace sources or categories, use a whitespace-separated list of patterns, optionally including wildcards (`*` and `?`).

Repeated events can only be suppressed if the `ArchMsg` trace source declares a key field.

`ArchMsgTrace` searches for the key field in the following way:

- It looks for the string `"\nPRIMARY KEY <key-field-name>"` in the description of the trace source and uses that field name if the string exists.
- If not found, it looks for a field named `"KEY"`.
- Otherwise, the `ArchMsg` trace source has no key field and cannot be suppressed.

If the `suppress_repeated` parameter is true, the plug-in suppresses repeated events for the same trace source that have the same key field value. For example, the key field might represent the PC and so repeated events for the same PC can be suppressed.



To see a list of all possible `ArchMsg` trace sources that the model can emit, run it with the `ListTraceSources` plug-in. Then search the output for trace sources with the `ArchMsg` prefix.

---

Some examples of `ArchMsg` trace sources are:

#### **`ArchMsg.Error.BusActiveDuringReset`**

A transaction was received at the bus slave port whilst reset was asserted.

#### **`ArchMsg.Warning.cache_contents_unknown`**

Execution that depends on unknown cache contents.

#### **`ArchMsg.Warning.warning_atomic_to_unsupported_memory`**

Atomic access to an unsupported memory type.

**ArchMsg.Warning.decode\_unpred\_other**

Use of unpredictable instruction.

**ArchMsg.Warning.recursive\_exception**

Recursive exception.

**Related information**

[ListTraceSources](#) on page 5189

### 4.3.1 ArchMsgTrace - parameters

This section describes the parameters for the ArchMsgTrace plug-in.

Each parameter is prefixed with `TRACE.ArchMsgTrace`, for example:

```
TRACE.ArchMsgTrace.exit_on_first_output
```

**Table 4-1: ArchMsgTrace parameters**

Name	Type	Default value	Allowed values	When set	Description
exit_on_first_output	bool	false	true, false	Init time	Exit the simulation process after the first message has been written.
filter_tags	string	"ALL"	""	Init time	Space-separated list of tags that are matched against the tag(s) of the trace events. The trace event message is printed if any of the tags matches. If the value is empty or ALL, all the messages are printed. Available tags: - ALL - UNPREDICTABLE - IMP_DEF.
suppress_categories	string	"Why"	""	Init time	Space-separated list of categories which should not be printed.
suppress_repeated	bool	true	true, false	Init time	Suppress repeated messages from similar call sites.
suppress_sources	string	""	""	Init time	Space-separated list of components or events that should not be printed.
trace-file	string	""	""	Init time	ArchMsgTrace output file.

## 4.4 ASTFplugin

ASTFplugin is an MTI plug-in that enables Fast Models to generate trace output in Architectural Structured Trace Format (ASTF).

ASTF is a binary, compressible trace format that captures the architectural execution of each of the CPUs in a system. It supports the collection of traces from complex workloads of up to billions of instructions in length. The format was designed to achieve a balance between compactness, ease of interpretation, and strong forwards and backwards compatibility.



ASTF and associated tools have been developed to support workload tracing, workload analysis, and to drive CPU performance models.

ASTFplugin can be used in combination with [ToggleMTIPlugin](#).



Note

- ASTFplugin is supported on Linux hosts only.
- ASTFplugin and the ASTF specification are in development and further iterations are expected. For the status of ASTFplugin, the version of the specification it supports, and any limitations and known issues, see the Fast Models release notes.

### Additional reading

- The ASTF specification is included in the Fast Models package in the `$FVLIB_HOME/Docs/` directory.
- For answers to some common queries about ASTFplugin, see the [ASTFplugin FAQs](#).
- For best practices for preparing an ASTF trace for performance prediction of a workload see [Workload Trace Generation Best Practices](#).
- For how to use a Fast Models FVP to capture an ASTF trace, see [How to generate ASTF traces of workloads running on Fast Models](#).

## 4.4.1 ASTFplugin - parameters

This section describes the parameters for the ASTFplugin plug-in.

Each parameter is prefixed with `TRACE.ASTFplugin`, for example:

```
TRACE.ASTFplugin.encoding-method
```

**Table 4-2: ASTFplugin parameters**

Name	Type	Default value	Allowed values	When set	Description
encoding-method	int	0x2	0x0 - 0x2	Runtime	ASTF record encoding method. 0: Uncompressed; 1: Compressed LZMA; 2: Compressed ZLib.
timestamp-enable	bool	true	true, false	Init time	Timestamp records will become part of the output if enabled.
timestamp-period	int	0x5	0x1 - 0x7fffffffffffffffff	Init time	This parameter sets the simulated time between two timestamps in micro-seconds.
trace-file	string	""	""	Runtime	Trace file pathname and prefix to write out to. Will be appended with component path, session number and .astf suffix.
verbosity	int	0x2	0x0 - 0x2	Runtime	Output verbosity level. 0: FATAL; 1: ERROR; 2: WARNING.

## 4.4.2 ASTFplugin usage notes

Be aware of the following points when using ASTFplugin.

- Load ASTFplugin in the same way as other plug-ins, using the syntax:

```
./isim_system <isim_params> --plugin /path/to/ASTFplugin.so <astf_plugin_params>
```

- ASTFplugin generates trace files with a `.astf` extension. During the simulation, these trace files might be incomplete. Incomplete trace files have a `.astf.part` extension and cannot be processed using the ASTF tools.
- The output trace files have a four digit enumerator field in the name. For example `FVP_Base_Cortex_A55.cluster0.cpu0.0000.astf`. This enumerator field is always present, regardless of whether ASTFplugin is used together with ToggleMTIPlugin. If ToggleMTIPlugin instructs ASTFplugin to stop and then resume, a new file is created for each CPU with each enumerator field incremented by one. However, if a CPU was not active when ToggleMTIPlugin instructed ASTFplugin to record, the respective output file is not created.
- ASTFplugin tries to register callbacks for MTI trace sources for the Scalable Vector Extension (SVE). If SVE is not enabled, ASTFplugin reports warnings to the console. You can ignore these warnings if SVE operations do not need to be recorded or if SVE is intended to be disabled.
- To improve performance, ASTFplugin is multithreaded. As the plug-in handles large streams of data, avoid using SMT or Hyper-Threading or running the threads on different sockets on a multi-socket host system. For optimal performance, we recommend you use `taskset` to restrict the model to using a specified set of N+1 host cores where N is the number of cores simulated in the model.

## 4.4.3 Additional ASTF support in Fast Models

In addition to ASTFplugin, Fast Models includes some other tools and libraries that support ASTF.

- The `trprint`, `trcheck`, `trdd`, `trimage`, and `trpidannotate` tools enable you to process the ASTF trace file. For details, see [ASTF tools](#).
- [ToggleMTIPlugin](#), installed in `$PVLIB_HOME/plugins/<OS_Compiler>/`, can be used together with ASTFplugin to limit trace generation to specific regions of interest.
- `libastf` library and header files:

### **`$PVLIB_HOME/astf_tools/lib/libastf.a`**

Library for reading and writing ASTF trace files. It exposes both C++ and C interfaces. For documentation, including a basic C++ example, see `$PVLIB_HOME/Docs/astf/libastf-api/libastf-api.txt`.

### **`$PVLIB_HOME/astf_tools/include/astf.h`**

Specifies the C++ and C interfaces to `libastf`. For documentation of each API function, see `$PVLIB_HOME/Docs/astf/libastf-api/<C++_function_name>.txt`.

### **`$PVLIB_HOME/astf_tools/include/astf_records.h`**

ASTF library record definitions.

- An example Python script, `$PVLIB_HOME/examples/pyIris/inst_count_trace_control.py`. It uses the `iris.debug` Python module to demonstrate using `ToggleMTIPlugin` to limit tracing to specific parts of the application. For usage instructions, run the script with the `-h` option. For more information, see the comments in the source file.

## Related information

[Iris Python Debug Scripting User Guide](#)

### 4.4.4 ASTF tools

The ASTF-related tools `trcheck`, `trdd`, `trimage`, `trpidannotate`, and `trprint` are installed in `$PVLIB_HOME/astf_tools/`. They enable you to process the trace files that `ASTFplugin` outputs, for example to view them in a human-readable format.

#### **trcheck**

Verifies the correctness of the trace files against the semantics defined in the ASTF format specification. For documentation, see `$PVLIB_HOME/Docs/astf/tools/trcheck.txt`.

#### **trdd**

Slices, copies, and (re)compresses the trace files. It can cut pieces from a trace file or re-encode a trace file by using a different compression level. For documentation, see `$PVLIB_HOME/Docs/astf/tools/trdd.txt`.

#### **trimage**

Analyses and profiles instructions and branches across multiple ASTF files. For documentation, see `$PVLIB_HOME/Docs/astf/tools/trimage.txt`.

#### **trpidannotate**

Annotates Context records in the trace files to correct the PID/TID information that was collected during tracing. For documentation, see `$PVLIB_HOME/Docs/astf/tools/trpidannotate.txt`.

#### **trprint**

Enables viewing and printing trace files in a human-readable format. For documentation, see `$PVLIB_HOME/Docs/astf/tools/trprint.txt`.

### 4.4.5 ASTFplugin FAQs

These FAQs answer some common queries about `ASTFplugin`.

#### **How do I trace a workload running in a multi-core Linux environment?**

I want to trace an application's workload running in a guest multi-core Linux environment but Linux OS migrates applications between cores. What should I do?

When using the `HLT` method of toggling trace with `ToggleMTIPlugin`, ASTF tracing in Fast Models is per-core. So, you must enable the `enable_trace_special_hlt_imm16` and `trace_special_hlt_imm16` parameters for all of the cores in the simulation.

See also [How to use ToggleMTIPlugin](#).

### What are the architectural limitations of ASTFplugin?

ASTF v0.11 supports up to Armv9.3-A. The plug-in is not guaranteed to work for architecture versions later than that.

### Does ASTF trace generation have a maximum trace file size?

No, it runs until your disk space is full and is killed by your OS.

### How much trace data should I capture?

A minimum trace log size of around 100M instructions is recommended to account for cache warming. A maximum of 10B instructions is recommended for efficient post-processing and analysis.

### If I generate traces for multiple programs, how do I distinguish between them in the trace logs?

If the guest Linux kernel is configured with 'CONFIG\_PID\_IN\_CONTEXTIDR' enabled, PID information is included in the `context` section of the trace logs. For example:

```
5 context : CPU in EL0, non-secure, thread-mode PID: 29193
```



PID information is only recorded if there is a PID change on a core.

### Can ASTFplugin trace process and thread IDs?

ASTFplugin supports recording PID/TID information through the `CONTEXTIDR_EL1` register, if the guest supports it. However that requires an additional pass for full PID/TID information. Is there a way to collect PID/TID information that doesn't require a re-run?

It is possible to do this with some additional post-processing. ASTFplugin traces PIDs using the `CONTEXTIDR_EL1` register and includes them in the trace, if they are available, in the first run. To fully match up the PIDs and TIDs in the trace, you then need to generate a PID-TID map from the OS and use the `trpidannotate` tool to amend the trace with the appropriate TID information.

See also [ASTF tools](#).

### Are timing controls, for example `cpi_div` and `cpi_mu1`, or Timing Annotation settings changeable at runtime?

No, they are only changeable at instantiation time.

### How does the plug-in distinguish between different clusters and cores?

The plug-in has no concept of the cluster and core topology of the model. It simply queries whether each component can execute code. If it can, the plug-in attaches itself to the trace sources

of the component that it needs to generate the ASTF streams. If not, it ignores that component. The resulting file names that include the clusters and cores are generated by the model and accepted by the plug-in.

### Why do files seem to be missing?

For example, in a model with two CPUs, I want to record two blocks. Files `cpu0.0000.astf` and `cpu1.0000.astf` record the first block and `cpu0.0001.astf` records the second block. Why is `cpu1.0001.astf` missing?

If a CPU is inactive while the plug-in is recording, the plug-in omits generating the associated file. Otherwise, the file would only contain the ASTF header and nothing else. So if a file is missing, it might be because that CPU was not active while the plug-in was recording.

### Why does the running counter for my files sometimes start with 0000 and sometimes with 0001?

When you request the plug-in to stop recording, it increments the running counter to ensure the next block's ASTF files have a different filename to those of the current block.

By default, the plug-in starts recording after it has initialised. However, if you request the plug-in to stop recording immediately, it increments the running counter and produces no `cpu*.0000.astf` files, because none of the CPUs had the chance to execute code. If you then request the plug-in to continue recording, the next block's stream files will have the 0001 counter value in their names.

### Why do I see messages about missing trace sources?

For example:

```
trace source SVE_LOADS/SVE_STORES not detected -> omitting registration
```

The plug-in is not feature-aware and therefore does not know if a CPU supports SVE. As a result, it always tries to register `SVE_LOADS` and `SVE_STORES` trace sources. If a CPU does not support SVE and this message appears, you can safely ignore it.

## 4.5 BranchPrediction

The `BranchPrediction` plug-in enables branch prediction modeling in Fast Models. The `BranchPrediction` plug-in is deprecated. It might be modified or removed in a future release.

Branch prediction is an aspect of Timing Annotation. For more details, see [Timing Annotation](#) in the Fast Models User Guide.

The type of branch predictor to use is selected using the `predictor-type` parameter. It can be one of the following example branch predictors that Arm provides, or a user-defined one:

#### FixedDirectionPredictor

Always takes a preset fixed direction.

**BiModalPredictor**

Standard 2-bit strength predictor.

**GSharePredictor**

Standard global history sharing predictor.

**HybridPredictor**

Selects the majority result from the bimodal predictor, fixed-direction predictor, and a random predictor.

**CortexA53Predictor**

Cortex®-A53 branch predictor. This is the default.

## 4.5.1 BranchPrediction - parameters

This section describes the parameters for the BranchPrediction plug-in.

Each parameter is prefixed with `BranchPrediction.BranchPrediction`, for example:

```
BranchPrediction.BranchPrediction.bpstat-branchcount
```

**Table 4-3: BranchPrediction parameters**

Name	Type	Default value	Allowed values	When set	Description
bpstat-branchcount	int	-0x1	0x0 - 0x7fffffffffffffffff	Init time	The number of branch instructions to display. Set to -1 to display all branch instructions.
bpstat-pathfilename	string	""	""	Init time	The path and filename of the branch statistics log.
mispredict-latency	int	0x8	0x0 - 0x7fffffffffffffffff	Init time	The number of instructions that are flushed for every misprediction.
predictor-type	string	"CortexA53Predictor"	""	Init time	The type of branch predictor to use.

## 4.5.2 BranchPrediction output example

This example command line configures and loads the BranchPrediction plug-in, using the FVP\_Base\_Cortex\_A73 example platform.

```
./isim_system -a __image.axf \
...
--plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/BranchPrediction.so \
-C BranchPrediction.BranchPrediction.predictor-type=BiModalPredictor \
-C BranchPrediction.BranchPrediction.mispredict-latency=8 \
-C BranchPrediction.BranchPrediction.bpstat-pathfilename=bpstat.txt \
-C BranchPrediction.BranchPrediction.bpstat-branchcount=5
```

This command produces the following log file, called `bpstat.txt`:

```
Processor Core: ARM_Cortex-A73
Cluster instance: 0
Core instance: 0
Mispredict Latency: 8
Image executed: _image.axf
PredictorType: BiModalPredictor
Total branch calls: 7757
Total Mispredictions: 130
Average prediction accuracy: 0.983241
Conditional Branches: 139
Total unique branch instructions: 289
--Branch instructions--
  PC Addr      Calls    Mispredict Accuracy
[0] 0x0         2         0          1
[1] 0x80000000  1         0          1
[2] 0x8000000c 10         0          1
[3] 0x80000014 10         0          1
[4] 0x8000001c 10         0          1
```

### 4.5.3 Other ways to report branch mispredictions

These are some alternative ways to report branch mispredictions.

- The `GenericTrace` plug-in can generate MTI trace events that report branch mispredictions. For example:

```
--plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/GenericTrace.so \
-C TRACE.GenericTrace.trace-sources=BRANCH_MISPREDICT
```

The `BranchPrediction` plug-in must also be loaded to generate these events.

- The `BranchPrediction` plug-in is integrated with the PMU event counters. On a Linux boot simulation, you can track the number of simulated branch mispredictions in an application by loading the `BranchPrediction` plug-in and running the `perf` tool. For example, the following command displays the number of branch mispredictions that are made in an application called `testapp`:

```
perf stat e branch-misses testapp
```

## 4.6 CADIIPCRemoteConnection

Use this plug-in to make a remote CADI connection to a model.

The remote connection can be from a client on a different host machine, or from a client on the same host but using a specific port.

To enable remote connections, set the `enable_remote_cadi` plug-in parameter to true and specify an IP address to listen to, or `0.0.0.0` to listen to all adapters. Also, you must start a CADI server with the `-s` model parameter.

The default values for this plug-in restrict connections to be from the localhost (127.0.0.1) only.

## 4.7 CDE

Custom Datapath Extension (CDE) allows you to improve performance and efficiency by adding application domain-specific features to embedded processors, while maintaining the advantages of the Arm® software ecosystem.

CDE allows you to add a customizable module inside some Cortex®-M processors. This module is driven by the pre-decoded CDE instructions and shares the same interface as the standard Arithmetic Logic Unit (ALU) of the processor.

Fast Models implements CDE using Model Trace Interface (MTI) plug-ins, with parameters to allow the plug-ins to be configured at runtime. The following Fast Models support CDE:

- [ARMCortexM33CT](#)
- [ARMCortexM52CT](#)
- [ARMCortexM55CT](#)
- [ARMCortexM85CT](#)
- [ARMAEMv8MCT](#)

The following model parameters are exposed for configuring CDE:

### **has\_cde**

Controls whether CDE is enabled. If enabled, a plug-in must be provided.

### **--plugin <path/to/plugin.so>**

This option can be specified multiple times, once for each CDE plug-in implementation.

Alternatively, plug-ins can be loaded by setting the `FM_PLUGINS` or `FM_TRACE_PLUGINS` environment variable.

### **cpu.cde\_impl\_name=<plugin\_name>**

The CDE implementation name to use with this core. If multiple CDE plug-in implementations are provided, each core can be requested to use a specific plug-in by using `cpu<n>.cde_impl_name=...`

Two example plug-ins are available, `CDETester` and `CDEConstant`. They are provided as pre-built libraries and as source code, located in `$PVLIB_HOME/plugins/source/`, to help with implementing your own plug-ins.



### 4.7.1 CDETester

CDETester is a basic example plug-in that allows you to specify at runtime which CDE instructions are supported by individual coprocessors and to specify the behavior, either nop or undefined.

To specify the instructions that coprocessors support, provide a plug-in parameter of the form:

```
CDE.CDETester.cde_tester_trivial.cps_implemented_instr=0xn
```

where `0xn` represents a hexadecimal bitmask of coprocessors that implement this instruction and `instr` represents a CDE instruction name.

The full list of CDE instruction names is as follows, where `a` represents dual variants and `v` represents vector variants:

- `cx1`
- `cx2`
- `cx3`
- `cx1d`
- `cx2d`
- `cx3d`
- `vcx1`
- `vcx2`
- `vcx3`
- `vcx1v`
- `vcx2v`
- `vcx3v`

Accumulate variants are handled in the same function implementation as the non-accumulate variants.

The bitmask takes the form:

**bits [7:0]**

For non-accumulate variants.

**bits [23:16]**

For accumulate variants.

The plug-in also allows control over which coprocessors implement CDE through the `CDE.CDETester.cde_tester_trivial.cps_implemented=0xn` parameter, where `0xn` represents a hexadecimal bitmask of coprocessors that implement CDE.

An example invocation to enable cx1 and cx1a (accumulate) for CP3 might be written as:

```
CDE.CDETester.cde_tester_trivial.cps_implemented=0x8 // enable CDE support for CP3
CDE.CDETester.cde_tester_trivial.cps_implemented_cx1=0x80008 // enable CX1 and CX1A
on CP3 (behaves as nop rather than undef)
```

On Linux, you can retrieve an up-to-date list of parameters from the model by using:

```
-l | grep -i cdetester
```

## 4.7.2 CDETester - parameters

This section describes the parameters for the CDETester plug-in.

Each parameter is prefixed with `CDE.CDETester`, for example:

```
CDE.CDETester.cde_tester_trivial.cps_implemented
```

**Table 4-4: CDETester parameters**

Name	Type	Default value	Allowed values	When set	Description
<code>cde_tester_trivial.cps_implemented</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Bitmask indicating coprocessors implemented.
<code>cde_tester_trivial.cps_implemented_cx1</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX1 instructions ([23:16] for CX1A, [7:0] for CX1).
<code>cde_tester_trivial.cps_implemented_cx1d</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX1D instructions ([23:16] for CX1DA, [7:0] for CX1D).
<code>cde_tester_trivial.cps_implemented_cx2</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX2 instructions ([23:16] for CX2A, [7:0] for CX2).
<code>cde_tester_trivial.cps_implemented_cx2d</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX2D instructions ([23:16] for CX2DA, [7:0] for CX2D).
<code>cde_tester_trivial.cps_implemented_cx3</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX3 instructions ([23:16] for CX3A, [7:0] for CX3).
<code>cde_tester_trivial.cps_implemented_cx3d</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX3D instructions ([23:16] for CX3DA, [7:0] for CX3D).
<code>cde_tester_trivial.cps_implemented_vcx1</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX1 instructions ([23:16] for VCX1A, [7:0] for VCX1).
<code>cde_tester_trivial.cps_implemented_vcx1v</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX1 (Vector) instructions ([23:16] for VCX1A (Vector), [7:0] for VCX1 (Vector)).
<code>cde_tester_trivial.cps_implemented_vcx2</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX2 instructions ([23:16] for VCX2A, [7:0] for VCX2).
<code>cde_tester_trivial.cps_implemented_vcx2v</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX2 (Vector) instructions ([23:16] for VCX2A (Vector), [7:0] for VCX2 (Vector)).
<code>cde_tester_trivial.cps_implemented_vcx3</code>	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX3 instructions ([23:16] for VCX3A, [7:0] for VCX3).

Name	Type	Default value	Allowed values	When set	Description
cde_tester_trivial.cps_implemented_vcx3v	int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX3 (Vector) instructions ([23:16] for VCX3A (Vector), [7:0] for VCX3 (Vector)).
has_cde_tester_trivial	bool	true	true, false	Init time	Whether the CDETester plugin is implemented (undefs all CDE instructions).

### 4.7.3 CDEConstant

This is an example plug-in that provides an implementation for every CDE instruction variant, for example [v]cxn, [v]cxnA, and [v]cxnD. Each instruction simply performs an XOR with some arguments and a constant.

To load this plug-in, use the following parameters when launching the model:

```
--plugin path/to/plugin/CDEConstant.so -C cpu.has_cde=1 -C
cpu.cde_impl_name=CDE_CONSTANT
```

### 4.7.4 CDEConstant - parameters

This section describes the parameters for the CDEConstant plug-in.

Each parameter is prefixed with `CDE.CDEConstant`, for example:

```
CDE.CDEConstant.has_cde_constant
```

**Table 4-5: CDEConstant parameters**

Name	Type	Default value	Allowed values	When set	Description
has_cde_constant	bool	true	true, false	Init time	Whether the CDEConstant plugin is implemented.

### 4.7.5 Implementing a CDE plug-in

Fast Models supports prototyping of custom instructions through a modular plug-in system, which uses the Model Trace Interface (MTI) framework.

Multiple CDE plug-ins can be registered with the model and each core can be instructed which plug-in behavior to use. Run-time configuration of CDE plug-ins is performed using plug-in parameters, although plug-in developers can use alternative approaches, for example configuration files.

This guide shows how to implement a basic MTI-based CDE plug-in, using the CDETester plug-in as an example. It is intended to be a quick start to plug-in development, and does not describe details about MTI. To learn more about MTI, see [Model Trace Interface Reference Manual](#). The source code for CDETester can be found in `$PVLIB_HOME/plugins/source/CDETester/`.

A CDE plug-in performs three main tasks:

- MTI and CDE interface registration.
- Handling parameters.
- Implementing CDE instructions.

### Related information

- [CDETester](#)

#### 4.7.5.1 Prerequisites for implementing a CDE plug-in

To build the CDE plug-in examples, you need the following:

- A compiler that matches the Fast Models build you are using.
- A recent version of CMake.
- An installation of the Fast Models package and libraries.

#### 4.7.5.2 CDE plug-in registration

After a CDE plug-in has been loaded into the model through the `--plugin` argument, or the `FM_TRACE_PLUGINS` environment variable, it must register itself with the CDE interface registry using the MTI framework API.

The CDE interface registry is responsible for:

- Managing all loaded CDE plug-ins.
- Passing any parsed arguments to the relevant CDE plug-in.
- Assigning CDE plug-ins to cores, as requested by the model arguments.

The CDE interface registry requires an interface name and version to be registered through MTI, as shown in `CDETester.cpp` and `CDETesterTrivialImpl.h`.

#### 4.7.5.3 CDE plug-in parameters

After the plug-in has registered itself with the CDE interface registry, the model passes any parsed command-line parameters to the CDE plug-in they are associated with.

The parameters to the `CDETester` example plug-in allow you to specify which custom instructions result in a no operation (nop) for each coprocessor. Any instructions that are not enabled result in an Undefined Instruction exception being raised.

The plug-in handles parameters that it receives from the model in `CDETester.cpp` by passing them through the `CDETesterFactory` interface to the handler implementation in `CDETrivialImpl::consumeParameter()`, defined in `CDETesterTrivialImpl.cpp`.

## Related information

- [CDETester](#)

### 4.7.5.4 CDE plug-in instruction handler interface

After plug-in registration and optional parameter handling, the plug-in should implement handlers for all available CDE instructions, even if it does not intend to implement custom functionality for all instruction variants.

To help you do this, Fast Models provides a utility header, `$PVLIB_HOME/include/ct/CDE/CDEInstHandlerInterface.h`. This header defines the `CDEInstHandlerInterface` class for handling CDE instruction calls, which plug-ins must inherit.

This interface declares pure virtual methods for each CDE instruction variant, for example dual and accumulator, as well as the structures containing decode information and call results. A typical signature follows this pattern:

```
CDEResult64 CDETrivialImpl::exec_cx2_d(const CX2DecodeInfo& decode_info, uint64_t
rfd_val, uint32_t rn_val)
```

The `CDETester` example plug-in inherits this interface in `CDETesterBaseImpl.h` and implements the instructions in `CDETesterTrivialImpl.h` and `CDETesterTrivialImpl.cpp`.

### 4.7.5.5 CDE plug-in instruction implementations

Each CDE instruction is mapped to a single function definition, except for accumulate variants, which are handled by checking for the accumulate flag in the parameters passed to the instruction implementations.

The full list of instructions that a plug-in is expected to implement is given in [CDETester](#).

Each instruction implementation accepts as parameters:

- A structure containing decoded instruction opcode parameters, including register numbers and immediate. For example `CX1DecodeInfo`.
- The contents of registers specified in the instruction opcode.

Instruction implementations should return a result structure of varying size, for example `CDEResult32` or `CDEResult64`. This structure indicates whether the instruction is supported, and if so, the return value and the number of cycles taken for execution, which is used in trace and performance analysis.

If a plug-in needs to raise an Undefined Instruction exception for a particular instruction, it can simply return a default-initialized result structure.

For full details of the expected function declarations and parameter types, see the file `$PVLIB_HOME/include/ct/CDE/CDEInstHandlerInterface.h`.

#### 4.7.5.6 Building the CDE plug-in

During plug-in development, either modify the example `cMakeLists.txt` files provided with the CDE plug-ins to reflect any changes to the file structure, or alternatively, use your own build system.

To build the example plug-ins:

1. Run `cmake` on the root directory of the plug-in to generate the project file, for example a Makefile or Visual Studio solution.
2. Run `make`.

### 4.7.6 CDE API

Reference documentation for the CDE API.

The CDE API is defined in the following header files, which are located in `$PVLIB_HOME/include/ct/CDE/`:

- `CDEFactoryInterface.h`
- `CDEInstHandlerInterface.h`
- `CDERegistryInterface.h`

#### 4.7.6.1 CDEFactoryInterface.h

Defines the interface for obtaining an instance of a `CDEInstHandlerInterface` for a specific core.

##### 4.7.6.1.1 CDE::CDEFactoryInterface class

The factory interface for obtaining an instance of a `CDEInstHandlerInterface` for a specific core.

##### 4.7.6.1.2 CDE::CDEFactoryInterface::instantiateCDEInstHandler()

Instantiate a `CDEInstHandlerInterface` for a given core. The caller owns the result.

```
virtual std::unique_ptr<CDEInstHandlerInterface>  
instantiateCDEInstHandler(std::string component_hierarchy) = 0;
```

#### **component\_hierarchy**

String representing the hierarchy of the current component.

#### 4.7.6.1.3 CDE::CDEFactoryInterface::CDEImplName()

Return the name of the CDE implementation. A core can use this method to disambiguate multiple CDE implementations in a simulation.

```
virtual std::string CDEImplName() const = 0;
```

#### 4.7.6.1.4 CDE::CDEFactoryInterface::CDEImplDescription()

Description of the CDE implementation.

Return the description of the CDE implementation that can be instantiated.

```
virtual std::string CDEImplDescription() const = 0;
```

#### 4.7.6.1.5 CDE::CDEFactoryInterface::CDEImplProviderName()

Provider name of the CDE implementation.

Return the name of the component providing the CDE implementation, for example a plug-in. This name might be used in informative diagnostic messages.

```
virtual std::string CDEImplProviderName() const = 0;
```

### 4.7.6.2 CDEInstHandlerInterface.h

Defines the interface for executing CDE instructions.

#### 4.7.6.2.1 CDE::CDEResult32 struct

32-bit result of a CDE instruction.

##### Members

##### **instr\_not\_supported**

Whether the instruction is supported.

##### **value**

Return value of the instruction.

##### **cycles**

Number of cycles of instruction execution.

#### 4.7.6.2.2 CDE::CDEResult64 struct

64-bit result of a CDE instruction.

##### Members

###### **instr\_not\_supported**

Whether the instruction is supported.

###### **value**

Return value of the instruction.

###### **cycles**

Number of cycles of instruction execution.

#### 4.7.6.2.3 CDE::CDEResult128 struct

128-bit result of a CDE instruction.

##### Members

###### **instr\_not\_supported**

Whether the instruction is supported.

###### **value\_lo**

Low 64 bits of the return value of the instruction.

###### **value\_hi**

High 64 bits of the return value of the instruction.

###### **cycles**

Number of cycles of instruction execution.

#### 4.7.6.2.4 CDE::CX1DecodeInfo struct

Decoded information for the cx1 instruction.

##### Members

###### **accumulate**

Whether to accumulate with existing register contents.

###### **coproc**

Number of coproc.

###### **imm**

Immediate value.

###### **rd\_num**

General-purpose destination register number.



#### 4.7.6.2.5 CDE::CX2DecodeInfo struct

Decoded information for the cx2 instruction.

##### Members

###### **accumulate**

Whether to accumulate with existing register contents.

###### **coproc**

Number of coproc.

###### **imm**

Immediate value.

###### **rd\_num**

General-purpose destination register number.

###### **rn\_num**

General-purpose source register number.

#### 4.7.6.2.6 CDE::CX3DecodeInfo struct

Decoded information for the cx3 instruction.

##### Members

###### **accumulate**

Whether to accumulate with existing register contents.

###### **coproc**

Number of coproc.

###### **imm**

Immediate value.

###### **rd\_num**

General-purpose destination register number.

###### **rn\_num**

General-purpose source register number.

###### **rm\_num**

General-purpose source register number.

#### 4.7.6.2.7 CDE::VCX1DecodeInfo struct

Decoded information for the vcx1 instruction.

##### Members

###### **accumulate**

Whether to accumulate with existing register contents.

###### **coproc**

Number of coproc.

###### **imm**

Immediate value.

###### **vd\_num**

Source and destination vector register number.

#### 4.7.6.2.8 CDE::VCX2DecodeInfo struct

Decoded information for the vcx2 instruction.

##### Members

###### **accumulate**

Whether to accumulate with existing register contents.

###### **coproc**

Number of coproc.

###### **imm**

Immediate value.

###### **vd\_num**

Source and destination vector register number.

###### **vm\_num**

Source vector register number.

#### 4.7.6.2.9 CDE::VCX3DecodeInfo struct

Decoded information for the vcx3 instruction.

##### Members

###### **accumulate**

Whether to accumulate with existing register contents.

###### **coproc**

Number of coproc.

**imm**

Immediate value.

**vd\_num**

Source and destination vector register number.

**vn\_num**

Source vector register number.

**vm\_num**

Source vector register number.

#### 4.7.6.2.10 [CDE::CDEInstHandlerInterface class](#)

Interface for executing CDE instructions.

This class defines the following methods for executing CDE instructions:

**exec\_cx1()**

cx1 instruction.

**exec\_cx1\_d()**

cx1D instruction.

**exec\_cx2()**

cx1 instruction.

**exec\_cx2\_d()**

cx2D instruction.

**exec\_cx3()**

cx3 instruction.

**exec\_cx3\_d()**

cx3D instruction.

**exec\_vcx\_1\_s()**

vcx1 instruction with S register.

**exec\_vcx\_1\_d()**

vcx1 instruction with D register.

**exec\_vcx\_1\_q()**

vcx1 instruction with Q register.

**exec\_vcx\_2\_s()**

vcx2 instruction with S register.

**exec\_vcx\_2\_d()**

vcx2 instruction with D register.

**exec\_vcx\_2\_q()**

vcx2 instruction with Q register.

**exec\_vcx\_3\_s()**

vcx3 instruction with S register.

**exec\_vcx\_3\_d()**

vcx3 instruction with D register.

**exec\_vcx\_3\_q()**

vcx3 instruction with Q register.

**exec\_vcx\_1\_beatwise()**

vcx1 instruction for one beat. Caller handles predicated writeback.

**exec\_vcx\_2\_beatwise()**

vcx2 instruction for one beat. Caller handles predicated writeback.

**exec\_vcx\_3\_beatwise()**

vcx3 instruction for one beat. Caller handles predicated writeback.

**4.7.6.2.11 CDE::CDEInstHandlerInterface::getCDECoprocessorMask()**

Return a bitmask indicating which coprocessor numbers this CDE implementation subsumes.

```
virtual uint8_t getCDECoprocessorMask() = 0;
```

**4.7.6.3 CDERegistryInterface.h**

Defines the interface to allow components, for instance plug-ins, to contribute CDE implementations to the simulation.

**4.7.6.3.1 CDE::CDERegistryInterface class**

Interface to register the CDE factory.

```
class CDERegistryInterface : public eslapi::CAInterface
```

This class is the interface to register the CDE factory into the Fast Models simulation component registry.

**4.7.6.3.2 CDE::CDERegistryInterface::registerCDEFactory()**

Register the CDE factory with the simulation.

```
virtual bool registerCDEFactory(std::ostream& error_stream, CDEFactoryInterface*  
interface) = 0;
```

**error\_stream**

The error stream.

**interface**

The CDE factory interface used to register.

#### 4.7.6.3.3 CDE::CDERegistryInterface::unregisterCDEFactory()

Unregister the CDE factory from the simulation.

```
virtual void unregisterCDEFactory(CDEFactoryInterface* interface) = 0;
```

**interface**

The CDE factory interface used to unregister.

#### 4.7.6.3.4 CDE::CDERegistryInterface::sendToCores()

Instantiate a CDEInstHandler and send it to the core.

The core can then call the CDE instruction execution functions provided by the plug-in on that CDEInstHandler.

```
virtual bool sendToCores() = 0;
```

## 4.8 CDELoader

CDELoader is a framework to enable rapid Arm Custom Instruction (ACI) prototyping.

It was introduced in Fast Models 11.27 as an alternative to the existing Custom Datapath Extension (CDE) plug-in framework.

To simplify the task of implementing Arm custom instructions, the CDELoader framework handles the required plug-in setup. The ACI developer only needs to provide a shared object, the ACI library.

The framework has the following parts:

**CDELoader**

A Fast Models plug-in designed to remove the need for MTI setup from ACI development. At runtime, CDELoader loads the shared object and executes the custom instruction implementation it provides.

CDELoader can accept multiple shared objects, allowing each core to specify which one to use. It also supports runtime configuration through an opaque string, which the ACI library can process according to its own requirements.

## API layer

A C99 API that defines the interface that the ACI library is bound by. It is located in `$PVLIB_HOME/include/ct/CDE/ACILibraryAPI.h`. It includes functions related to:

- Library creation, deletion, naming, and versioning.
- Custom instruction execution. These are the `aci_exec_*` functions.
- [Custom instruction mnemonics](#).

## ACI library

A shared object that provides implementations for the custom instructions and library functions.

As the CDELoader framework uses a C99 API, the library can be written in C or in another language that can produce ABI-compatible shared objects and can interface with C. For example, C++, Rust, Go, or Python, through C extensions.



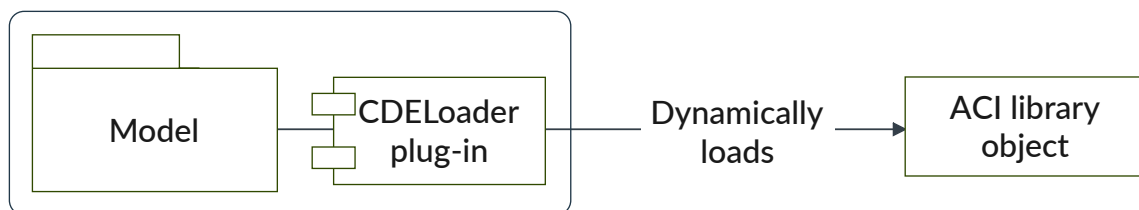
Note

Languages other than C can require extra wrapper code to ensure proper interaction with the C99 API and to meet the shared object requirements. The implementation and compatibility details might vary depending on the language and toolchain used.

An example ACI library, called ACIConstant, is provided as both a pre-built library and as source code, located in `$PVLIB_HOME/plugins/source/ACIConstant/` to help you implement your own libraries. This plug-in mirrors the example [CDEConstant](#) implementation for the [CDE](#) plug-in framework.

This figure shows the framework:

**Figure 4-1: CDELoader and ACI library**



The following model parameters are exposed for using CDELoader:

**-C `cpu.has_cde=[0|1]`**

Controls whether this core enables CDE.

**-C `cpu.cde_impl_name=<aci_library_name>`**

Sets the ACI library name to use with this core. The library name is returned by `aci_get_library_name()`.

If multiple shared objects are provided, each core can request to use a specific implementation by using `cpu<n>.cde_impl_name=...`

**--plugin <path/to/CDELoader.so>**

Loads the CDELoader plug-in. Alternatively, set the `FM_PLUGINS` environment variable to the full path of the plug-in. For more details, see [Loading a plug-in](#).

For the CDELoader plug-in parameters, see [CDELoader - parameters](#).

For example:

```
./isim_system \
-C cpu_has_cde=1 \
--plugin $plugin_dir/CDELoader.so \
-C cpu.cde_impl_name=ACI_LIBRARY_NAME \ # defined in aci_get_library_name()
-C CDE.CDELoader.aci_object_files="path/to/aci_library_obj_file, path/to/another/
aci_library_obj_file" \
-C CDE.CDELoader.aci_parameters="Opaque string parameter to be processed by the ACI
library"
```

## 4.8.1 CDELoader - parameters

This section describes the parameters for the CDELoader plug-in.

Each parameter is prefixed with `CDE.CDELoader`, for example:

```
CDE.CDELoader.aci_object_files
```

**Table 4-6: CDELoader parameters**

Name	Type	Default value	Allowed values	When set	Description
aci_object_files	string	""	""	Init time	Comma-separated list of path to shared objects to load.
aci_parameters	string	""	""	Init time	Opaque string parameter to be processed by the ACI library.

## 4.8.2 ACI library implementation

Each CDE instruction is mapped to a single function definition, except for accumulate variants which are handled by checking for the accumulate flag in the parameters passed to the instruction implementations.

The following table lists the CDE instruction functions that an ACI library must implement. A *d* suffix represents a dual variant and a *v* suffix represents a vector variant.

**Table 4-7: Mapping CDE instructions to ACI library function definitions**

Instruction	ACILibraryAPI.h function
cx1	aci_exec_cx1()
cx2	aci_exec_cx2()

Instruction	ACILibraryAPI.h function
cx3	aci_exec_cx3()
cx1d	aci_exec_cx1_d()
cx2d	aci_exec_cx2_d()
cx3d	aci_exec_cx3_d()
vcx1 (single-register variant)	aci_exec_vx1_s()
vcx2 (single-register variant)	aci_exec_vx2_s()
vcx3 (single-register variant)	aci_exec_vx3_s()
vcx1 (double-register variant)	aci_exec_vx1_d()
vcx2 (double-register variant)	aci_exec_vx2_d()
vcx3 (double-register variant)	aci_exec_vx3_d()
vcx1v	aci_exec_vx1_beatwise()
vcx2v	aci_exec_vx2_beatwise()
vcx3v	aci_exec_vx3_beatwise()



The library must also implement some library-related functions. For full details of the library function definitions, see `$PVLIB_HOME/include/ct/CDE/ACILibraryAPI.h`.

If you do not wish to support a particular instruction or instruction variant, the function implementation can simply return `ACI_STATUS_NOT_IMPLEMENTED`.

The ACI API defines an opaque pointer, `ACIHandle`, which, for complex implementations, can be mapped to an object of a class that handles the library functions. For simple and quick implementations, you can ignore the handle pointer.

### ACI library dependencies

CDELoader dynamically loads shared objects that are standalone and do not depend on external libraries, for example compiler-specific libraries.

To resolve the external dependencies, either include them statically when compiling the shared object or ensure they exist in a path that is visible to the model. Always list all dependencies for your shared objects and check they are present before running the model.

### 4.8.3 ACIConstant example ACI library

You can find the source code for an example ACI library which implements `ACILibraryAPI.h` in `$PVLIB_HOME/plugins/source/ACIConstant/ACILibrary.cpp`.

The implementation mirrors the output of the [CDEConstant](#) plug-in.



The example is also provided as a pre-built library. To load this library, add the following parameters when launching the model:

```
--plugin $plugin_dir/CDELoader.so \  
-C cpu.has_cde=1 \  
-C cpu.cde_impl_name=ACI_CONSTANT \  
-C CDE.CDELoader.aci_object_files=$plugin_dir/ACIConstant.so
```

Alternatively, you can build the library yourself using the example Makefile provided with the ACIConstant source code. If you have problems loading the library while running the model, ensure all necessary compiler-specific libraries are visible to the model.

## 4.8.4 Custom instruction mnemonics

This is an optional feature that allows you to specify meaningful, human-readable mnemonics for your custom instructions to improve the readability of log files.

`ACILibraryAPI.h` declares a struct named `ACIMnemonics` which contains the strings used as custom mnemonics. These strings can be retrieved using the `aci_get_custom_mnemonics()` function, and they can subsequently be used by trace plug-ins loaded with the model.

Refer to the [ACIConstant example ACI library](#) source code for an example on how to define custom mnemonics.

## 4.8.5 ACI library API

Reference documentation for the ACI library API.

The ACI library API is defined in the header file `ACILibraryAPI.h`, which is located at `$PVLIB_HOME/include/ct/CDE/ACILibraryAPI.h`.

### 4.8.5.1 ACIHandle

Opaque type specifying a handle for the ACI library. This handle is passed to all the `aci_exec_*` functions.

```
typedef struct ACIHandleInstance* ACIHandle;
```

#### 4.8.5.2 aci\_get\_library\_version()

Return the library version, ACI\_API\_VERSION.

This is part of a check conducted by the model to confirm that both it and the library are operating on the same version of this API.

```
ACI_EXPORT uint16_t aci_get_library_version(void);
```

#### 4.8.5.3 aci\_get\_library\_name()

Return the library name.

The model parameter `cpu.cde_impl_name` is used to select the library. This parameter can be used to quickly change which library is used when providing multiple libraries. Each core can request to use a specific library by using `cpu<n>.cde_impl_name=...`

```
ACI_EXPORT const char* aci_get_library_name(void);
```

#### 4.8.5.4 aci\_get\_coprocessor\_mask()

Provide a bitmask that signifies the coprocessor numbers encompassed by this ACI implementation.

```
ACI_EXPORT uint8_t aci_get_coprocessor_mask(void);
```

#### 4.8.5.5 aci\_set\_param()

Sets the parameter provided to the model for the ACI library.

```
ACI_EXPORT void aci_set_param(const char* parameter);
```

##### **parameter**

A generic string that can be processed in any way that fits the library's implementation. This can be set using the model parameter `CDELoader.aci_parameters`.

#### 4.8.5.6 aci\_new()

Constructor of the ACI library.

Return a handle object which can be nullptr if there is no handle needed.

```
ACI_EXPORT ACIHandle aci_new(void);
```

### 4.8.5.7 aci\_free()

Destructor of the ACI library.

```
ACI_EXPORT void aci_free(ACIHandle handle);
```

**handle**

Object to free, can be ignored if there was no handle allocated.

### 4.8.5.8 ACICX1DecodeInfo struct

Decode information of cx1 instruction.

**Members****accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**rd\_num**

General-purpose destination register number.

### 4.8.5.9 aci\_exec\_cx1()

Instruction CX1.

Return ACI\_STATUS\_OK on success, or ACI\_STATUS\_NOT\_IMPLEMENTED if instruction is not implemented.

```
ACI_EXPORT ACI_Status aci_exec_cx1(ACIHandle handle,
                                   const ACICX1DecodeInfo* decode_info,
                                   uint32_t rd_val,
                                   uint32_t* result);
```

**handle**

ACI handle object created by aci\_new().

**decode\_info**

Decoded fields for the CX1 instruction.

**rd\_val**

Value of the destination register.

**result**

Pointer to the value returned by the CX1 instruction.

#### 4.8.5.10 `aci_exec_cx1_d()`

Instruction CX1D.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_cx1_d(ACIHandle handle,
                                   const ACICX1DecodeInfo* decode_info,
                                   uint64_t rfd_val,
                                   uint64_t* result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the CX1D instruction.

**rfd\_val**

Value of the first of the destination register pair.

**result**

Pointer to the value returned by the CX1D instruction.

#### 4.8.5.11 `ACICX2DecodeInfo` struct

Decode information of cx2 instruction.

### Members

**accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**rd\_num**

General-purpose destination register number.

**rn\_num**

General-purpose source register number.

#### 4.8.5.12 aci\_exec\_cx2()

Instruction CX2.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_cx2(ACIHandle      handle,
                                   const ACICX2DecodeInfo* decode_info,
                                   uint32_t         rd_val,
                                   uint32_t         rn_val,
                                   uint32_t*        result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the CX2 instruction.

**rd\_val**

Value of the destination register.

**rn\_val**

Value of the source register.

**result**

Pointer to the value returned by the CX2 instruction.

#### 4.8.5.13 aci\_exec\_cx2\_d()

Instruction CX2D.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_cx2_d(ACIHandle      handle,
                                      const ACICX2DecodeInfo* decode_info,
                                      uint64_t         rfd_val,
                                      uint32_t         rn_val,
                                      uint64_t*        result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the CX2D instruction.

**rfd\_val**

Value of the first of the destination register pair.

**rn\_val**

Value of the source register.

**result**

Pointer to the value returned by the CX2D instruction.

#### 4.8.5.14 ACICX3DecodeInfo struct

Decode information of cx3 instruction.

##### Members

**accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**rd\_num**

General-purpose destination register number.

**rn\_num**

General-purpose source register number.

**rm\_num**

General-purpose source register number.

#### 4.8.5.15 aci\_exec\_cx3()

Instruction CX3.

Return `ACI_STATUS_OK` ON SUCCESS, OR `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_cx3(ACIHandle      handle,
                                   const ACICX3DecodeInfo* decode_info,
                                   uint32_t         rd_val,
                                   uint32_t         rn_val,
                                   uint32_t         rm_val,
                                   uint32_t*         result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the CX3 instruction.

**rd\_val**

Value of the destination register.

**rn\_val**

Value of the source register.

**rm\_val**

Value of the source register.

**result**

Pointer to the value returned by the CX3 instruction.

#### 4.8.5.16 `aci_exec_cx3_d()`

Instruction CX3D.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_cx3_d(ACIHandle handle,
                                     const ACICX3DecodeInfo* decode_info,
                                     uint64_t rfd_val,
                                     uint32_t rn_val,
                                     uint32_t rm_val,
                                     uint64_t* result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the CX3D instruction.

**rfd\_val**

Value of the first of destination register pair.

**rn\_val**

Value of the source register.

**rm\_val**

Value of the source register.

**result**

Pointer to the value returned by the CX3D instruction.

#### 4.8.5.17 `ACIVCX1DecodeInfo` struct

Decoded information for the vcx1 instruction.

### Members

**accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**vd\_num**

Source and destination vector register number.

**4.8.5.18 aci\_exec\_vcx1\_s()**

Instruction VCX1 with Single-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx1_s(ACIHandle      handle,
                                     const ACIVCX1DecodeInfo* decode_info,
                                     uint32_t          sd_val,
                                     uint32_t*         result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX1 instruction.

**sd\_val**

32-bit value of the floating-point source and destination register.

**result**

Pointer to the value returned by the VCX1 instruction.

**4.8.5.19 aci\_exec\_vcx1\_d()**

Instruction VCX1 with Double-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx1_d(ACIHandle      handle,
                                     const ACIVCX1DecodeInfo* decode_info,
                                     uint64_t          dd_val,
                                     uint64_t*         result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX1 instruction.

**dd\_val**

64-bit value of the floating-point source and destination register.

**result**

Pointer to the value returned by the VCX1 instruction.



#### 4.8.5.20 aci\_exec\_vcx1\_beatwise()

Instruction VCX1 for one beat. Caller handles predicated writeback.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx1_beatwise(ACIHandle          handle,
                                             const ACIVCX1DecodeInfo* decode_info,
                                             uint32_t          d_val,
                                             uint8_t           beat,
                                             uint8_t           elmt_mask,
                                             uint32_t*         result);
```

##### **handle**

ACI handle object created by `aci_new()`.

##### **decode\_info**

Decoded fields for the VCX1 instruction.

##### **d\_val**

32-bit value of the source and destination vector register.

##### **beat**

Beat-number to index into the Q regs.

##### **elmt\_mask**

Element mask.

##### **result**

Pointer to the value returned by the VCX1 instruction.

#### 4.8.5.21 ACIVCX2DecodeInfo struct

Decoded information for the vcx2 instruction.

##### **Members**

##### **accumulate**

Whether to accumulate with existing register contents.

##### **coproc**

Number of coproc.

##### **imm**

Immediate value.

##### **vd\_num**

Source and destination vector register number.

##### **vm\_num**

Source vector register number.

#### 4.8.5.22 aci\_exec\_vcx2\_s()

Instruction VCX2 with Single-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx2_s(ACIHandle          handle,
                                     const ACIVCX2DecodeInfo* decode_info,
                                     uint32_t             sd_val,
                                     uint32_t             sm_val,
                                     uint32_t*             result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX2 instruction.

**sd\_val**

32-bit value of the floating-point source and destination register.

**sm\_val**

32-bit value of the floating-point source register.

**result**

Pointer to the value returned by the VCX2 instruction.

#### 4.8.5.23 aci\_exec\_vcx2\_d()

Instruction VCX2 with Double-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx2_d(ACIHandle          handle,
                                     const ACIVCX2DecodeInfo* decode_info,
                                     uint64_t             dd_val,
                                     uint64_t             dm_val,
                                     uint64_t*             result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX2 instruction.

**dd\_val**

64-bit value of the floating-point source and destination register.

**dm\_val**

64-bit value of the floating-point source register.

**result**

Pointer to the value returned by the VCX2 instruction.

#### 4.8.5.24 `aci_exec_vcx2_beatwise()`

Instruction VCX2 for one beat. Caller handles predicated writeback.

Return `ACI_STATUS_OK` on success, `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx2_beatwise(ACIHandle handle,
                                             const ACIVCX2DecodeInfo* decode_info,
                                             uint32_t d_val,
                                             uint32_t m_val,
                                             uint8_t beat,
                                             uint8_t elmt_mask,
                                             uint32_t* result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX2 instruction.

**d\_val**

32-bit value of the source and destination vector register.

**m\_val**

32-bit value of the source vector register.

**beat**

Beat-number to index into the Q regs.

**elmt\_mask**

Element mask.

**result**

Pointer to the value returned by the VCX2 instruction.

#### 4.8.5.25 `ACIVCX3DecodeInfo` struct

Decoded information for the vcx3 instruction.

### Members

**accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**vd\_num**

Source and destination vector register number.

**vn\_num**

Source vector register number.

**vm\_num**

Source vector register number.

#### 4.8.5.26 `aci_exec_vcx3_s()`

Instruction VCX3 with Single-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx3_s(ACIHandle          handle,
                                     const ACIVCX3DecodeInfo* decode_info,
                                     uint32_t             sd_val,
                                     uint32_t             sn_val,
                                     uint32_t             sm_val,
                                     uint32_t*            result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX3 instruction.

**sd\_val**

32-bit value of the floating-point source and destination register.

**sn\_val**

32-bit value of the floating-point source register.

**sm\_val**

32-bit value of the floating-point source register.

**result**

Pointer to the value returned by the VCX3 instruction.

#### 4.8.5.27 `aci_exec_vcx3_d()`

Instruction VCX3 with Double-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx3_d(ACIHandle          handle,
                                     const ACIVCX3DecodeInfo* decode_info,
```

```
uint64_t dd_val,
uint64_t dn_val,
uint64_t dm_val,
uint64_t* result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX3 instruction.

**dd\_val**

64-bit value of the floating-point source and destination register.

**dn\_val**

64-bit value of the floating-point source register.

**dm\_val**

64-bit value of the floating-point source register.

**result**

Pointer to the value returned by the VCX3 instruction.

#### 4.8.5.28 `aci_exec_vcx3_beatwise()`

Instruction VCX3 for one beat. Caller handles predicated writeback.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx3_beatwise(ACIHandle handle,
const ACIVCX3DecodeInfo* decode_info,
uint32_t d_val,
uint32_t n_val,
uint32_t m_val,
uint8_t beat,
uint8_t elmt_mask,
uint32_t* result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX3 instruction.

**d\_val**

32-bit value of the source and destination vector register.

**n\_val**

32-bit value of the source vector register.

**m\_val**

32-bit value of the source vector register.

**beat**

Beat-number to index into the Q regs.

**elmt\_mask**

Element mask.

**result**

Pointer to the value returned by the VCX3 instruction.

#### 4.8.5.29 ACIMnemonics struct

Optional custom mnemonics for the custom instructions.

##### Members

**cx1**

Instruction CX1 mnemonic.

**cx1a**

Instruction CX1 (Accumulator variant) mnemonic.

**cx1d**

Instruction CX1D mnemonic.

**cx1da**

Instruction CX1D (Accumulator variant) mnemonic.

**cx2**

Instruction CX2 mnemonic.

**cx2a**

Instruction CX2 (Accumulator variant) mnemonic.

**cx2d**

Instruction CX2D mnemonic.

**cx2da**

Instruction CX2D (Accumulator variant) mnemonic.

**cx3**

Instruction CX3 mnemonic.

**cx3a**

Instruction CX3 (Accumulator variant) mnemonic.

**cx3d**

Instruction CX3D mnemonic.

**cx3da**

Instruction CX3D (Accumulator variant) mnemonic.

**vcx1\_s**

Instruction VCX1 (Single-register variant) mnemonic.

**vcx1a\_s**

Instruction VCX1 (Single-register accumulator variant) mnemonic.

**vcx1\_d**

Instruction VCX1 (Double-register variant) mnemonic.

**vcx1a\_d**

Instruction VCX1 (Double-register accumulator variant) mnemonic.

**vcx1\_v**

Instruction VCX1 vector mnemonic.

**vcx1a\_v**

Instruction VCX1 vector (Accumulator variant) mnemonic.

**vcx2\_s**

Instruction VCX2 (Single-register variant) mnemonic.

**vcx2a\_s**

Instruction VCX2 (Single-register accumulator variant) mnemonic.

**vcx2\_d**

Instruction VCX2 (Double-register variant) mnemonic.

**vcx2a\_d**

Instruction VCX2 (Double-register accumulator variant) mnemonic.

**vcx2\_v**

Instruction VCX2 vector mnemonic.

**vcx2a\_v**

Instruction VCX2 vector (Accumulator variant) mnemonic.

**vcx3\_s**

Instruction VCX3 (Single-register variant) mnemonic.

**vcx3a\_s**

Instruction VCX3 (Single-register accumulator variant) mnemonic.

**vcx3\_d**

Instruction VCX3 (Double-register variant) mnemonic.

**vcx3a\_d**

Instruction VCX3 (Double-register accumulator variant) mnemonic.

**vcx3\_v**

Instruction VCX3 vector mnemonic.

**vcx3a\_v**

Instruction VCX3 vector (Accumulator variant) mnemonic.

## 4.9 Crypto

The `crypto` plug-in enables Armv8 and Armv9 processor models to support the Armv8.0 Cryptographic Extensions and Armv8.3 architected Pointer Authentication algorithms.

The `crypto` plug-in is available for download from the [Arm Developer website](#).

When the plug-in is loaded:

- All Armv8 and Armv9 processors in the system implement all functionality from the Armv8.0 Cryptographic Extensions by default, although you can disable it for specific processors by setting their `CRYPTODISABLE` parameter.
- All Armv8.3 and Armv9 processors in the system implement the architected algorithms for Pointer Authentication and Generic Authentication by default. Plug-in parameters control which processors in the system have architected algorithms enabled.

AEMs, for example, AEMvACT, have parameters that allow you to restrict the `crypto` plug-in features. These parameters use the same encodings as the flags within the AArch32 `ID_ISAR5` and AArch64 `ID_AA64ISAR0_EL1` system registers. You can set these parameters for a specific AEM core using this syntax:

```
-C cpu.cpu<X>.<feature_name>=<value>
```

Where `feature_name` can be one of the following:

- `crypto_aes`, with these possible values:
  - 0**  
No AES instructions are implemented
  - 1**  
The `AESE`, `AESD`, `AESMC`, and `AESIMC` instructions are implemented
  - 2**  
As 1, but in addition, the `PMULL` and `PMULL2` instructions can operate on 64-bit data values. This is the default value.
- `crypto_sha1`, with these possible values:
  - 0**  
No SHA-1 instructions are implemented
  - 1**  
The `SHA1C`, `SHA1P`, `SHA1M`, `SHA1H`, `SHA1SU0`, and `SHA1SU1` instructions are implemented. This is the default value.
- `crypto_sha256`, with these possible values:
  - 0**  
No SHA-256 instructions are implemented
  - 1**  
The `SHA256H`, `SHA256H2`, `SHA256SU0`, and `SHA256SU1` instructions are implemented. This is the default value.



- `crypto_sha3`, with these possible values:
  - 0**  
No Armv8.4 SHA-3 instructions are implemented. This is the default value.
  - 1**  
SHA-3 instructions are implemented if Armv8.4 is enabled.
  - 2**  
SHA-3 instructions are implemented.
- `crypto_sha512`, with these possible values:
  - 0**  
No Armv8.4 SHA-512 instructions are implemented. This is the default value.
  - 1**  
SHA-512 instructions are implemented if Armv8.4 is enabled.
  - 2**  
SHA-512 instructions are implemented.
- `crypto_sm3`, with these possible values:
  - 0**  
No Armv8.4 SM-3 instructions are implemented. This is the default value.
  - 1**  
SM-3 instructions are implemented if Armv8.4 is enabled.
  - 2**  
SM-3 instructions are implemented.
- `crypto_sm4`, with these possible values:
  - 0**  
No Armv8.4 SM-4 instructions are implemented. This is the default value.
  - 1**  
SM-4 instructions are implemented if Armv8.4 is enabled.
  - 2**  
SM-4 instructions are implemented.

For example, to disable the AES instructions on core 0:

```
./isim_system --plugin Crypto.so -C cpu.cpu0.crypto_aes=0
```



These parameters are only available for AEMs. For other Armv8-A and Armv9-A models, the behavior is fixed to the default values.

---

## 4.9.1 Crypto - parameters

This section describes the parameters for the Crypto plug-in.

Each parameter is prefixed with `CRYPTO.Crypto`, for example:

```
CRYPTO.Crypto.authentication_algorithm
```

**Table 4-8: Crypto parameters**

Name	Type	Default value	Allowed values	When set	Description
<code>authentication_algorithm</code>	string	"QARMA5"	""	Init time	Choice of PACAlgorithm. Valid values: "QARMA5", "QARMA3". "QARMA3" can be enabled only if the core feature <code>has_qarma3_pac</code> is true. Default value: "QARMA5". The pattern is a comma separated list of glob-style patterns. These patterns are matched against the instance name of cores in the system. If a pattern matches then the Architected algorithm is installed on the core if that core supports ARMv8.3. For example: - "*", install the algorithm on all supported cores - "cluster0.*", install the algorithm on all cores in cluster0 - "cluster0.cpu0" install the algorithm only on cluster0.cpu0 - "cluster0.*,cluster1.cpu1" install the algorithm on all cores in cluster0 and also cluster1.cpu1 - "" (an empty string), do not install the algorithm on any core .
<code>generic_authentication_core_pattern</code>	string	"*"	""	Init time	install the ARMv8.3 Architected Generic Authentication algorithm only on ARMv8.3 cores matching one of these patterns. The pattern is a comma separated list of glob-style patterns. These patterns are matched against the instance name of cores in the system. If a pattern matches then the Architected algorithm is installed on the core if that core supports ARMv8.3. For example: - "*", install the algorithm on all supported cores - "cluster0.*", install the algorithm on all cores in cluster0 - "cluster0.cpu0" install the algorithm only on cluster0.cpu0 - "cluster0.*,cluster1.cpu1" install the algorithm on all cores in cluster0 and also cluster1.cpu1 - "" (an empty string), do not install the algorithm on any core .
<code>pointer_authentication_core_pattern</code>	string	"*"	""	Init time	install the ARMv8.3 Architected Pointer Authentication algorithm only on ARMv8.3 cores matching one of these patterns. The pattern is a comma separated list of glob-style patterns. These patterns are matched against the instance name of cores in the system. If a pattern matches then the Architected algorithm is installed on the core if that core supports ARMv8.3. For example: - "*", install the algorithm on all supported cores - "cluster0.*", install the algorithm on all cores in cluster0 - "cluster0.cpu0" install the algorithm only on cluster0.cpu0 - "cluster0.*,cluster1.cpu1" install the algorithm on all cores in cluster0 and also cluster1.cpu1 - "" (an empty string), do not install the algorithm on any core .
<code>verbose</code>	int	0x0	0x0 - 0x1	Init time	verbosity level. 0, terse. 1, verbose.

## 4.10 GDBRemoteConnection

The `GDBRemoteConnection` plug-in allows the model to be debugged using GDB. It is included in the Third Party IP add-on package to Fast Models.

For more details about the add-on package, contact [Arm Technical Support](#).

Load the plug-in using the following command-line options:

- `--allow-debug-plugin`, or the short version, `-D`
- `--plugin`

For example:

```
./isim_system --allow-debug-plugin --plugin GDBRemoteConnection.so
```

Then, a suitable GDB can be connected to the model using the GDB `target` command.

`GDBRemoteConnection` supports the following operations:

- Connection to models that contain a single core cluster.
- Read and write of core registers.
- Read and write of memory.
- Run, stop, single step.
- Breakpoints.
- Connection to AArch32 and AArch64 models.

### 4.10.1 GDBRemoteConnection limitations

This section describes the limitations of the `GDBRemoteConnection` plug-in.

- Connections are only allowed to single-core simulations, not to multiprocessor or multicluster simulations.
- No tracepoint support.
- No parameter support.
- The memory view only shows the current memory space.
- The disassembly only uses the current instruction set.
- Breakpoints can only be set on the current memory space.

You cannot use GDB when debugging software that uses semihosting. When a program tries to use semihosting with the GDB plug-in, the GDB debugger wrongly reports having hit a breakpoint:

```
Program received signal SIGTRAP, Trace/breakpoint trap.
```

When using the `GDBRemoteConnection` plug-in, start the simulation with the `-D` (or `-s`) flag. Attaching the `GDBRemoteConnection` plug-in to a running simulation can cause segmentation faults on the simulation and the GDB client.

## 4.11 GenericCounter

GenericCounter is an example MTI plug-in that prints to stdout the number of occurrences of a specific trace source when the simulation terminates.

GenericCounter only counts a single trace source, which is set using the `TRACE.GenericCounter.trace-source` parameter. To count multiple trace sources, load the plug-in multiple times. In this case, each plug-in instance has a unique name which you use instead of GenericCounter when setting its parameters. The names are either set implicitly or explicitly. This example sets them explicitly as `counter1` and `counter2`:

```
--plugin counter1=path/to/GenericCounter.so \
--plugin counter2=path/to/GenericCounter.so \
-C TRACE.counter1.trace-source=EXCEPTION \
-C TRACE.counter2.trace-source=READ_ACCESS
```

Otherwise, each plug-in instance has an implicit name which consists of the plug-in name and a sequential suffix, for example `GenericCounter`, `GenericCounter0`, `GenericCounter1` and so on. This example uses the implicit names:

```
--plugin path/to/GenericCounter.so --plugin path/to/GenericCounter.so \
-C TRACE.GenericCounter.trace-source=EXCEPTION \
-C TRACE.GenericCounter0.trace-source=READ_ACCESS
```

The source code for this plug-in is provided in `$PVLIB_HOME/examples/MTI/GenericCounter/`.

### 4.11.1 GenericCounter - parameters

This section describes the parameters for the GenericCounter plug-in.

Each parameter is prefixed with `TRACE.GenericCounter`, for example:

```
TRACE.GenericCounter.print_on_event
```

**Table 4-9: GenericCounter parameters**

Name	Type	Default value	Allowed values	When set	Description
<code>print_on_event</code>	string	""	""	Init time	If set, print the count information to stdout when <code>print_on_event</code> trace source fires. If empty, only print the count information at the end of the simulation or when the <code>print_stats</code> parameter is written to.
<code>print_stats</code>	int	0x0	0x0 - 0x0	Runtime	On write, print count information to stdout.
<code>trace-source</code>	string	"INST"	""	Init time	The trace source to be counted. Example: <code>BRA_DIR</code> .

## 4.12 GenericTrace

GenericTrace is a flexible MTI plug-in that allows you to configure which events are traced, using a comma-separated list of trace sources. Output can be printed to a file or to the terminal.

Specify one or more trace sources using the `trace-sources` parameter, for example:

```
./isim_system --plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/GenericTrace.so -C
TRACE.GenericTrace.trace-sources=EXCEPTION,EXCEPTION_RETURN
```



To see a list of the available trace sources for each component in the model that provides trace, run the model with the `ListTraceSources` plug-in. See [ListTraceSources](#) for details.

The `trace-sources` parameter provides flexibility:

- To specify trace sources that match a pattern, use the `*` or `?` wildcards, for example:

```
TRACE.GenericTrace.trace-sources=EXCEPTION*
```

- To collect a trace source for a specific component only, prepend the trace source with the hierarchical path of the component, for example:

```
TRACE.GenericTrace.trace-sources=FVP_Base_AEMvA.cluster0.cpu0.EXCEPTION
```

You can optionally use wildcards in either or both of the names of the trace component and the trace source, for example:

```
TRACE.GenericTrace.trace-sources=FVP_Base_AEMvA.cluster0.*.EXCEPTION*
```

- To trace specific fields in a trace source, append a field mask. For example to trace only the second field (`pc`) of the `INST` trace source, use:

```
TRACE.GenericTrace.trace-sources=INST=0x2
```

Or, to filter out the 8th field (`ELEMENT_SIZE`) from the `CORE_STORES` trace source, use:

```
TRACE.GenericTrace.trace-sources=CORE_STORES=0xFFF7F
```

- If no trace sources are specified, `GenericTrace` by default traces all the instructions.

The source code for this plug-in is provided as a programming example in `$PVLIB_HOME/examples/MTI/GenericTrace/source/`.



This plug-in can be used with [ToggleMTIPlugin](#).

### 4.12.1 GenericTrace - parameters

This section describes the parameters for the GenericTrace plug-in.

Each parameter is prefixed with `TRACE.GenericTrace`, for example:

```
TRACE.GenericTrace.collect-latest-data-only
```

**Table 4-10: GenericTrace parameters**

Name	Type	Default value	Allowed values	When set	Description
collect-latest-data-only	bool	false	true, false	Init time	Collect only latest N trace source fires and print upon destruction.
enabled	bool	true	true, false	Runtime	If set to true, tracing is enabled.
flush	bool	false	true, false	Runtime	If set to true, the trace file is flushed after every event. This has a performance impact but could be used to better debug crashes.
hide-fieldnames	bool	false	true, false	Runtime	Do not print field names when printing trace output.
latest-data-size	int	0x1	0x0 - 0x7fffffffffffffff	Init time	Size of latest data to store in buffer for capturing only the latest trace source fires.
mti_enum_wrap_value_in_quotes	bool	false	true, false	Init time	Enclose MTI_ENUM values in quotes.
perf-period	int	0x0	0x0 - 0x7fffffffffffffff	Init time	Print performance information every N instructions. 0 means disabled.
print-timestamp	bool	false	true, false	Runtime	Start each trace entry with the host time.
shorten-paths	bool	true	true, false	Runtime	If set to true, the component paths of trace events are shortened by removing the common prefix. The minimal, non-ambiguous path suffix remains. If all traced sources belong to the same components, no path is logged. Default is true.
simulated-timestamp	bool	false	true, false	Runtime	Start each trace entry with the simulated time.
start-icount	int	0x0	0x0 - 0x7fffffffffffffff	Init time	Start tracing on a certain instruction count. Tracing starts up to 2048 instructions before this count.

Name	Type	Default value	Allowed values	When set	Description
stop-icount	int	0x7fffffffffffffff	0x0 - 0x7fffffffffffffff	Init time	Stop tracing on a certain instruction count. Tracing stops up to 2048 instructions after this count.
stop_on_event	bool	false	true, false	Init time	Stop the simulation when any event is triggered.
trace-file	string	""	""	Runtime	The trace file to write into. If STDERR, prints to stderr. If empty, prints to stdout.
trace-file-limit	int	0x0	0x0 - 0x7fffffffffffffff	Init time	The limit of the size of the output file in bytes. The simulation is stopped when this size is reached. If 0, it is unlimited.
trace-sources	string	"INST"	""	Runtime	A comma-separated list of trace sources to be traced. A component path can be prepended, with components separated by dots. Both the component path and the trace source name can contain the wildcards * and ?. A field mask as a number in hex or decimal format can be appended with =. Example: my.subsystem.core.cpu*.TRACE_SOURCE=0x08.
verbose	bool	false	true, false	Runtime	Print some debugging information.

## 4.12.2 GenericTrace plug-in usage example

This example shows how to use the GenericTrace plug-in to trace accesses by the graphics driver to the registers of a GPU register model.

### Procedure

1. Use the ListTraceSources plug-in to list the trace sources that the platform provides. It is located in \$PVLIB\_HOME/plugins/<OS\_compiler>/:

```
${PATH_Model} --plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/ListTraceSources.so
```

The plug-in prints the following information to the terminal:

- All components that provide trace, including the GPU model, for example:

```
Component (292) providing trace: Kits3_Subsys.css.gpu
```

- The trace sources provided by the GPU model. For example, these are some generic trace sources provided by Mali™ GPU models:

#### **INFO\_ReadRegister**

Access time, addresses, data, and names of the registers that were read.

#### **INFO\_Reset**

GPU reset data.

#### **INFO\_WriteRegister**

Access time, addresses, and names of the registers that were updated, and the data before and after the update.

**INFO\_Irq**

Name and state of the IRQ signal from the GPU. The name indicates the type of IRQ signal; GPU, Job Manager, or MMU. The state can be Y for Set, or N for Clear.

**WARN\_ReadToWriteOnlyRegister**

Warning messages and addresses for the write-only registers that have been read by the graphics driver.

**WARN\_WriteToReadOnlyRegister**

Warning messages and addresses for the read-only registers that have been written by the graphics driver.

**WARN\_AccessToUnimplementedRegister**

Warning messages and addresses for the invalid registers that have been accessed by the graphics driver.

2. To trace all events from the GPU model, launch the platform with the following additional options:

```
--plugin $PVLIB_HOME/plugins/Linux64 GCC-9.3/GenericTrace.so \
-C TRACE.GenericTrace.trace-sources=Kits3_Subsys.css.gpu.* \
-C TRACE.GenericTrace.enabled=1 \
-C TRACE.GenericTrace.verbose=1 \
-C TRACE.GenericTrace.print-timestamp=1 \
-C TRACE.GenericTrace.trace-file=dp-trace-generic.log
```

Where:

- Kits3\_Subsys.css.gpu is the GPU model listed in Step 1.
- Kits3\_Subsys.css.gpu.\* means trace all the GPU-supported trace sources. Alternatively, (not shown in this example):
  - To trace one specific GPU trace source only, add it as a suffix to the GPU model. For instance, Kits3\_Subsys.css.gpu.INFO\_ReadRegister.
  - To trace multiple specific trace sources, use a comma-separated list. For instance, Kits3\_Subsys.css.gpu.INFO\_ReadRegister, Kits3\_Subsys.css.gpu.INFO\_WriteRegister.
- The trace-file option specifies the log file in which to save the trace output. If it is not used, the trace results are shown on the host terminal.

## Results

The host terminal or the log file shows details about the driver-accessed registers, such as the register addresses, data, and the access time. For example:

```
HOST_TIME=1557460.545195s INFO_ReadRegister: REG_OFFSET=0x0000000000000000
VALUE=0x60000000 REG_NAME="GPU_ID"
HOST_TIME=1557460.545266s INFO_ReadRegister: REG_OFFSET=0x0000000000000004
VALUE=0x07130206 REG_NAME="L2_FEATURES"
HOST_TIME=1557460.545279s INFO_ReadRegister: REG_OFFSET=0x0000000000000008
VALUE=0x00000000 REG_NAME="SUSPEND_SIZE"
HOST_TIME=1557460.545291s INFO_ReadRegister: REG_OFFSET=0x000000000000000c
VALUE=0x00000809 REG_NAME="TILER_FEATURES"
HOST_TIME=1557460.545303s INFO_ReadRegister: REG_OFFSET=0x0000000000000010
VALUE=0x00000001 REG_NAME="MEM_FEATURES"
```



```

HOST_TIME=1557460.545316s INFO_ReadRegister: REG_OFFSET=0x0000000000000014
VALUE=0x00002830 REG_NAME="MMU_FEATURES"
HOST_TIME=1557460.545325s INFO_ReadRegister: REG_OFFSET=0x0000000000000018
VALUE=0x000000ff REG_NAME="AS_PRESENT"
HOST_TIME=1557460.545334s INFO_ReadRegister: REG_OFFSET=0x000000000000001c
VALUE=0x00000007 REG_NAME="JS_PRESENT"
HOST_TIME=1557460.545345s INFO_ReadRegister: REG_OFFSET=0x00000000000000c0
VALUE=0x0000020e REG_NAME="JS0_FEATURES"
HOST_TIME=1557460.545362s INFO_ReadRegister: REG_OFFSET=0x00000000000000c4
VALUE=0x000001fe REG_NAME="JS1_FEATURES"
HOST_TIME=1557460.545364s INFO_ReadRegister: REG_OFFSET=0x00000000000000c8
VALUE=0x0000007e REG_NAME="JS2_FEATURES"
HOST_TIME=1515565849.690948s gpu.INFO_WriteRegister:
REG_OFFSET=0x00000000000001870 VALUE=0x00000000 UPDATED_VALUE=0x00000000
REG_NAME="JOB_SLOT0_JS_FLUSH_ID_NEXT"
HOST_TIME=1515565849.691304s gpu.INFO_WriteRegister:
REG_OFFSET=0x00000000000001860 VALUE=0x00000000 UPDATED_VALUE=0x00000001
REG_NAME="JOB_SLOT0_JS_COMMAND_NEXT"
HOST_TIME=1515565849.691322s gpu.INFO_Irq: IRQ_NAME="IRQ_JOB" IRQ_STATE=Y
HOST_TIME=1515565849.691561s gpu.INFO_ReadRegister:
REG_OFFSET=0x0000000000000100c VALUE=0x00000001 REG_NAME="JOB_IRQ_STATUS"
HOST_TIME=1515565849.691643s gpu.INFO_WriteRegister:
REG_OFFSET=0x00000000000001004 VALUE=0x00000000 UPDATED_VALUE=0x00000001
REG_NAME="JOB_IRQ_CLEAR"
HOST_TIME=1515565849.691647s gpu.INFO_Irq: IRQ_NAME="IRQ_JOB" IRQ_STATE=N

```

### 4.12.3 Mapping between SYSREG\_UPDATE trace sources and SPSR registers

For tracing updates to SPSR\_\* registers, GenericTrace maps the fields in the registers to fields in SYSREG\_UPDATE32 or SYSREG\_UPDATE64 trace sources.

The mapping is shown in the following table:

**Table 4-11: Mapping between SYSREG\_UPDATE\* trace sources and register encodings for SPSR\_\* registers**

SYSREG_UPDATE32 or SYSREG_UPDATE64 field	Register field
opc0	R
opc	M
CRn	M1
CRm	O
opc2	O

## 4.13 ListTraceSources

ListTraceSources is an MTI plug-in that displays a complete and self-documenting list of the trace sources that a model provides, without running the model.

The plug-in prints output for each component in the model, either to stdout or to a file. For example:

```
...
Component (4) providing trace: FVP_VE_Cortex_A7x1.cluster.cpu0 (ARM_Cortex-A7,
11.4.60)
=====
Component is of type "ARM_Cortex-A7"
Version is "11.4.60"
#Sources: 195

Source ASYNC_MEMORY_FAULT (Context ID Register write.)
  Field FAULT type:MTI_UNSIGNED_INT size:4 (Fault status in ESR format)
  Field VADDR type:MTI_SIGNED_INT size:8 (Virtual Address (or 0 if unavailable))
  Field PADDR type:MTI_UNSIGNED_INT size:8 (Physical Address (or 0 if
unavailable))
...

```

The source code for this plug-in is also provided as a programming example in `$PVLIB_HOME/examples/MTI/ListTraceSources/source/`.

### 4.13.1 ListTraceSources - parameters

This section describes the parameters for the ListTraceSources plug-in.

Each parameter is prefixed with `TRACE.ListTraceSources`, for example:

```
TRACE.ListTraceSources.file
```

**Table 4-12: ListTraceSources parameters**

Name	Type	Default value	Allowed values	When set	Description
file	string	""	""	Init time	File to write the list of trace sources to. Default is to write to the console.
print_components_only	bool	false	true, false	Init time	If true, the plug-in prints the trace component information only, not the sources or fields.

## 4.14 PipelineModel

Use the `PipelineModel` plug-in to estimate the performance of workloads within a Fast Models environment.

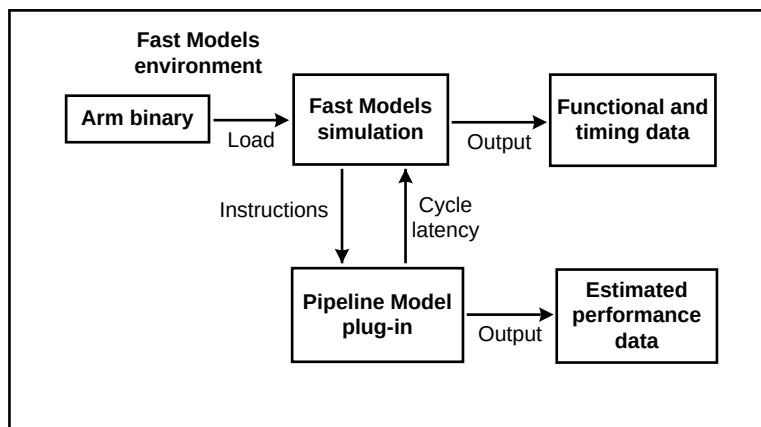


The `PipelineModel` plug-in is deprecated. It might be modified or removed in a future release.

The plug-in models first-order effects of microarchitecture components on the overall Cycles Per Instruction (CPI) value. Examples of such effects are data and structural hazards due to instruction dependencies.

The `PipelineModel` is implemented as a Fast Models plug-in. It processes instruction traces and injects cycle latencies into the simulation. The plug-in is non-intrusive to the functional accuracy of the simulation.

**Figure 4-2: Overview of PipelineModel plug-in**



Fast Models provides the following prebuilt example PipelineModel plug-ins:

### CortexA53PipelineModel

An approximation of the performance characteristics of the in-order, dual issue properties of the Cortex®-A53 processor. The model provides estimated performance characteristics of a given compute-bound workload in terms of Cycles Per Instruction.



The model should not be used as a reference for hardware performance as it has limitations, such as the absence of a cache model.

## InOrderPipelineModel

An implementation of a single-issue 4-stage pipelined processor that illustrates the basic components of a CPU. It demonstrates how components such as Fetch, Decode, and Execute can be implemented as a Fast Models plug-in. The source code is provided in `$PVLIB_HOME/plugins/source/PipelineModel/Cores/InOrder/`. It contains a README and a makefile for building the example.



Note

The model does not represent any Arm® core and is intended only as a guide for developing more advanced PipelineModels using Fast Models.

The `PipelineModel` is an aspect of Timing Annotation. For more details, see [Timing Annotation](#) in the Fast Models User Guide.

### 4.14.1 CortexA53PipelineModel - parameters

This section describes the parameters for the `CortexA53PipelineModel` plug-in.

Each parameter is prefixed with `PipelineModel.CortexA53PipelineModel`, for example:

```
PipelineModel.CortexA53PipelineModel.core-type
```

**Table 4-13: CortexA53PipelineModel parameters**

Name	Type	Default value	Allowed values	When set	Description
core-type	string	""	""	Init time	Core type description.
end-pc	string	"0x00000000"	""	Init time	End processing instructions after this pc.
instance-name	string	""	""	Init time	Cluster and core num description.
output	string	""	""	Init time	Defaults to stdout. Specify STDERR or a filename.
start-pc	string	"0x00000000"	""	Init time	Start processing instructions from this pc.

### 4.14.2 InOrderPipelineModel - parameters

This section describes the parameters for the `InOrderPipelineModel` plug-in.

Each parameter is prefixed with `PipelineModel.InOrderPipelineModel`, for example:

```
PipelineModel.InOrderPipelineModel.core-type
```

**Table 4-14: InOrderPipelineModel parameters**

Name	Type	Default value	Allowed values	When set	Description
core-type	string	""	""	Init time	Core type description.

Name	Type	Default value	Allowed values	When set	Description
instance-name	string	""	""	Init time	Cluster and core num description.

### 4.14.3 PipelineModel example

This example shows how the `PipelineModel` plug-in generates data. This example uses a single issue, 6-stage pipeline. The stages are Fetch, Decode, Issue, EX1, EX2, and WR.

		0	1	2	3	4	5	6	7	8	9	10
[0]	ADD R1, R2, R3	IF	ID	IS	X0	X1	WR					
[1]	MUL R4, R1, R2		IF	ID	IS	IS*	X0	X1	WR			
[2]	ADD R5, R4, R6			IF	ID	ID*	IS	IS*	X0	X1	WR	
[3]	ADD R7, R8, R9				IF	IF*	ID	ID*	IS	X0	X1	WR

In this instruction sequence, pipeline stalls are shown with an asterisk. The following stalls occur:

- Instruction [1] stalls one cycle in the Issue stage until R1 has been written.
- Instruction [2] stalls one cycle in the Decode stage due to a structural hazard because the Issuer is still in use by Instruction [1].
- Instruction [2] stalls one cycle in the Issue stage until R4 has been written.
- Instruction [3] stalls one cycle in the Fetch stage due to a structural hazard because the Decoder is still in use by Instruction [2].
- Instruction [3] stalls one cycle in the Decode stage due to a structural hazard because the Issuer is still in use by Instruction [2].

The `PipelineModel` plug-in uses the accumulated stalls for each instruction to calculate the runtime latency as follows:

- Instruction [0] latency = 0 cycles.
- Instruction [1] latency = 1 cycle.
- Instruction [2] latency = 2 cycles.
- Instruction [3] latency = 2 cycles.

In this example, there are five stall cycles in total. At the end of the simulation, the `PipelineModel` plug-in uses the accumulated stalls to produce a total cycle count. It uses the cycle count to determine the final Cycles Per Instruction (CPI) or Instructions Per Cycle (IPC) value.

#### 4.14.4 Naming the plug-in instance

You can optionally assign a name to the plug-in instance. This is useful in a multiprocessor platform if you load the same plug-in multiple times. The assigned name is used to identify which plug-in instance the parameters apply to.

For example, the following commands load the `PipelineModel` plug-in twice and assign the names `CortexA53_0` and `CortexA53_1` to the first and second instances respectively:

```
--plugin CortexA53_0=CortexA53PipelineModel.so
--plugin CortexA53_1=CortexA53PipelineModel.so
```

You can then specify the plug-in instance name in the plug-in parameters. For example:

```
-C PipelineModel.CortexA53_0.core-type=ARM_Cortex-A53
-C PipelineModel.CortexA53_0.instance-name=Base.cluster0.cpu0

-C PipelineModel.CortexA53_1.core-type=ARM_Cortex-A53
-C PipelineModel.CortexA53_1.instance-name=Base.cluster0.cpu1
```

#### 4.14.5 Example command lines

The following command lines show how to load the `PipelineModel` plug-in in a variety of configurations. Any parameters that are not relevant to the examples have been omitted. All examples use the `FVP_Base_Cortex_A53` example platform unless otherwise stated.

##### Default usage, with no options

The following command line attaches the plug-in to any instance and sends output to `stdout`:

```
./isim_system
...
--plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/CortexA53PipelineModel.so
```

##### Single cluster, single processor platform

The following command line attaches the plug-in to `FVP_Base_Cortex_A53.cluster0.cpu0`. It outputs the file `stat.txt`:

```
./isim_system
...
-C cluster0.NUM_CORES=1
--plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/CortexA53PipelineModel.so
-C PipelineModel.CortexA53PipelineModel.core-type=ARM_Cortex-A53
-C PipelineModel.CortexA53PipelineModel.instance-
name=FVP_Base_Cortex_A53.cluster0.cpu0
-C PipelineModel.CortexA53PipelineModel.output=stat.txt
```

## Single cluster, multiprocessor platform, default usage

The following command line attaches the plug-in to all Cortex®-A53 processors:

```
./isim_system
...
-C cluster0.NUM_CORES=4
--plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/CortexA53PipelineModel.so
```

## Single cluster, multiprocessor platform

The following command line loads two plug-in instances in a dual-processor platform. The first plug-in instance is named `cortexA53_0` and is attached to `FVP_Base_Cortex_A53.cluster0.cpu0`. The second plug-in instance is named `cortexA53_1` and is attached to `FVP_Base_Cortex_A53.cluster0.cpu1`:

```
./isim_system
...
-C cluster0.NUM_CORES=2
--plugin CortexA53_0=$PVLIB_HOME/plugins/Linux64_GCC-9.3/CortexA53PipelineModel.so
--plugin CortexA53_1=$PVLIB_HOME/plugins/Linux64_GCC-9.3/CortexA53PipelineModel.so
-C PipelineModel.CortexA53_0.core-type=ARM_Cortex-A53
-C PipelineModel.CortexA53_0.instance-name=FVP_Base_Cortex_A53.cluster0.cpu0
-C PipelineModel.CortexA53_1.core-type=ARM_Cortex-A53
-C PipelineModel.CortexA53_1.instance-name=FVP_Base_Cortex_A53.cluster0.cpu1
```

## Multicluster, single processor platform

The following command line uses the `FVP_Base_Cortex_A73x1_A53x1` example platform. It names the plug-in instance `core1` and attaches it to `FVP_Base_Cortex_A73x1_A53x1.cluster1.cpu0`:

```
./isim_system
...
--plugin Core1=$PVLIB_HOME/plugins/Linux64_GCC-9.3/CortexA53PipelineModel.so
-C PipelineModel.Core1.core-type=ARM_Cortex-A53
-C PipelineModel.Core1.instance-name=FVP_Base_Cortex_A73x1_A53x1.cluster1.cpu0
```

## Plug-in core-type mismatch

The following command line specifies a Cortex®-A55 processor. This mismatches the platform, which is `FVP_Base_Cortex_A53`. The plug-in fails to load:

```
./isim_system
...
--plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/CortexA53PipelineModel.so
-C PipelineModel.CortexA53PipelineModel.core-type=ARM_Cortex-A55
```

## Plug-in instance-name mismatch

The following command line specifies an `instance-name` that does not exist. The plug-in fails to load:

```
./isim_system
-C cluster0.NUM_CORES=1
...
--plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/CortexA53PipelineModel.so
-C PipelineModel.CortexA53PipelineModel.core-type=ARM_Cortex-A53
```

```
-C PipelineModel.CortexA53PipelineModel.instance-
name=FVP_Base_Cortex_A53.cluster0.cpu1
```

## Range processing

The following command-line arguments set an address range. The `PipelineModel` only processes instructions within this range:

```
...
-C PipelineModel.CortexA53PipelineModel.start-pc=0x80000001
-C PipelineModel.CortexA53PipelineModel.end-pc=0x8000FFFF
```

If the start and end of the range are identical, the `PipelineModel` processes all instructions, which is the same as not specifying a range at all.

## Cycle limit

To run the `PipelineModel` with a cycle limit, use the model option `--cyclelimit`. When the cycle limit is reached, the simulation terminates and the output file is generated.

The following command line arguments causes the `PipelineModel` to process all instructions until the cycle limit is reached:

```
...
--cyclelimit=30000000
--plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/CortexA53PipelineModel.so
```



Note

The `--cyclelimit` option starts counting from the instruction at the beginning of the simulation, not from the instruction at `start-pc`. If the cycle limit is reached before the `start-pc` is hit, the `PipelineModel` does not process any instructions.

## 4.14.6 PipelineModel output

At the end of the simulation, the `PipelineModel` generates a Cycles Per Instruction (CPI) value along with other performance data. You can use this for further analysis.

For example:

```
Elapsed time: 8 seconds
Instructions per second: 484186
Simulated CPU speed: 1.484186 MHz
CPU cycles: 5159984
RAW stalls: 3551018
Instructions issued: 3873492
Instructions retired: 3873491
Loads executed: 921403
Stores executed: 923306
IPC: 0.750679
CPI: 1.33213
```



The CPI value is the primary metric that measures the performance of workloads. It is calculated using the formula:

$$\text{CPI} = \text{cycles elapsed} / \text{instructions retired}$$

The lower the CPI, the better the performance. As a general indication, a CPI of 0.5 on a dual-issue, in-order or out-of-order processor means that an instruction takes 0.5 cycles to complete. In this case, the pipeline units are maximized and no latencies are generated.

Conversely, performance can be measured in Instructions Per Cycle (IPC), where:

$$\text{IPC} = \text{instructions retired} / \text{cycles elapsed}$$

The higher the IPC, the better the performance. As a general indication, an IPC of two on a dual issue, in-order or out-of-order processor means that on average, two instructions commit in each cycle.

## 4.15 RunTimeParameterTest

`RunTimeParameterTest` is an example MTI plug-in that demonstrates how to add new string, integer, and boolean parameters at runtime.

This plug-in is provided only as source code, in `$PVLIB_HOME/examples/MTI/RunTimeParameterTest/source/`.

## 4.16 Sidechannel

The Sidechannel plug-in allows communication between the software on the host and software on the target. It is no longer used.

## 4.17 TarmacText

`TarmacText` is an MTI plug-in that extracts the architectural execution trace, also known as Tarmac, of a processor. `TarmacText` extracts the trace in a textual form and saves it in a file.



`TarmacText` is deprecated. We recommend you only use it if you specifically require the `TarmacText` trace format. Otherwise, use the `TarmacTrace` plug-in instead.

---

The plug-in allows you to trace multiple components simultaneously, saving the generated traces in different files.

Enable trace generation by setting the `component` parameter to the required component name or space-separated names for multiple components. The default value of `component` is empty, which means the plug-in finds and traces all active processors.

Output filenames are composed of a common prefix, configurable with the `log` parameter, followed by the name of the component, and terminated with the extension `.log`. The default value of the prefix is `tarmac`.



Note

The platform name is trimmed from the component name.

### 4.17.1 TarmacText - parameters

This section describes the parameters for the TarmacText plug-in.

Each parameter is prefixed with `TarmacText.`, for example:

```
TarmacText.component
```

**Table 4-15: TarmacText parameters**

Name	Type	Default value	Allowed values	When set	Description
<code>component</code>	string	<code>"*"</code>	""	Init time	A space-separated list of components to trace. Supports globbing (see <code>man 7 glob</code> ). Does not restrict nor extend the list of components for which the ExecStep Iris EventSource will be published.
<code>evs</code>	string	""	""	Init time	Filename to log the binary tarmac into. Supports the substitution pattern <code>@COMP@</code> .
<code>exec</code>	string	""	""	Init time	Shell command to execute to process the binary tarmac on the fly. Supports the substitution pattern <code>@COMP@</code> . The command must accept on its standard input the tarmac event stream.
<code>flush</code>	bool	<code>false</code>	<code>true</code> , <code>false</code>	Init time	Whether to flush the output files specified by the <code>log</code> parameter as often as possible. Decreases the performance.
<code>log</code>	string	<code>"tarmac.@COMP@.log"</code>	""	Init time	Filename to log the tarmac text into. Supports the substitution pattern <code>@COMP@</code> .
<code>start</code>	int	<code>0</code>	<code>≥0</code>	Init time	The amount of cycles as defined by the PERIODIC event to wait for before starting to trace. This can be used to reduce the impact of the tracing on performance until the portion of interest is reached. Does not impact the ExecStep Iris EventSource.

## 4.18 TarmacTrace

TarmacTrace is an MTI plug-in that prints Tarmac trace to `stdout` or to a file. This section describes the format of the output.

The trace might include instructions executed, program flow, updates to registers, memory accesses made by Arm® cores in the simulation, and other information. Plug-in parameters control the amount and type of information that is traced.



This plug-in can be used with [ToggleMTIPlugin](#).

### 4.18.1 TarmacTrace - parameters

This section describes the parameters for the TarmacTrace plug-in.

Each parameter is prefixed with `TRACE.TarmacTrace`, for example:

```
TRACE.TarmacTrace.end-instruction-count
```

**Table 4-16: TarmacTrace parameters**

Name	Type	Default value	Allowed values	When set	Description
end-instruction-count	int	0x0	0x0 - 0x7fffffffffffffffff	Init time	The instruction count when the tracing should be stopped. Default is to never stop tracing.
instruction-count-is-per-target	bool	true	true, false	Init time	If true (default) then the start-instruction-count and end-instruction-count parameters apply to individual targets separately. If false, all components start and stop tracing at once when the first component reaches the instruction count.
loadstore-display-width	int	0x8	0x0 - 0x40	Init time	Memory transactions can involve up to 64 bytes. For easier readability these can be broken up into multiple memory access records with a smaller number of bytes. 0 means do not break up any transaction.
quantum-size	int	0x2710	0x1 - 0x7fffffffff	Init time	Set the default quantum size used to compute when the tracing should start and stop, in instructions. This is overridden by the <code>CORE_INFO.QUANTUM_SIZE</code> trace source field of the component, if present.
quiet	bool	false	true, false	Init time	Limit output to trace information.
start-instruction-count	int	0x0	0x0 - 0x7fffffffffffffffff	Init time	The instruction count when the tracing should start. Default is to trace from the beginning.

Name	Type	Default value	Allowed values	When set	Description
trace-file	string	""	""	Runtime	Trace output file. The default is an empty string, which means stdout unless MTI_TARMAC_LOG is set. STDOUT means stdout. STDERR means stderr. Setting this parameter at runtime causes the current trace file to be flushed and closed and a new one to be opened. Writing at runtime, STDOUT, STDERR, and MTI_TARMAC_LOG are not supported when trace-file-per-comp=1.
trace-file-per-comp	bool	false	true, false	Init time	Write trace to multiple files.
trace-inst-stem	string	""	""	Init time	Base instance path to select a group of instances to trace.
trace_aarch64_vfp_full_width	bool	false	true, false	Init time	Trace a write to an S or D register in AArch64 as a write to the corresponding V register.
trace_atomic	bool	true	true, false	Init time	Trace memory update by atomic operation.
trace_branches	bool	false	true, false	Init time	Trace changes of the program flow like direct or indirect branches and exception returns.
trace_bte	bool	true	true, false	Init time	Trace opcode rejected by BTE.
trace_bus_accesses	bool	false	true, false	Init time	Trace bus accesses by the core. This includes accesses by the caches of the core. This considerably slows down the model.
trace_cache	bool	true	true, false	Init time	Trace cache fills and evictions.
trace_core_registers	bool	true	true, false	Init time	Trace the core registers R0-R14, the CPSR, and the SPSR registers.
trace_cp15	bool	true	true, false	Init time	Trace writes to CP15 registers.
trace_dap	bool	true	true, false	Init time	Trace accesses on the debug access port.
trace_ete	bool	true	true, false	Init time	Trace packets generated by the ETE.
trace_events	bool	true	true, false	Init time	Trace events, for example exceptions and mode changes.
trace_exception_reasons	bool	true	true, false	Init time	Trace INFO_EXCEPTION_REASONS (M-class only so far).
trace_generic_events	bool	false	true, false	Init time	Trace generic events.
trace_gic	bool	true	true, false	Init time	Trace GIC register writes and updates.
trace_gic_reads	bool	false	true, false	Init time	Trace GIC register reads.
trace_gic_signal_changes	bool	false	true, false	Init time	Trace GIC FIQ/IRQ Signal Changes.
trace_gic_state_change	bool	false	true, false	Init time	Trace GIC interrupt state changes.
trace_gic_table_walks	bool	false	true, false	Init time	Trace GIC table walks.
trace_gicv3	bool	true	true, false	Init time	Trace GICv3 memory mapped accesses.
trace_gicv3_comms	int	0x0	0x0 - 0x7	Init time	Trace GICv3 communications between cores and distributor. Bitfield; 1 = trace CPU; 2 = trace RD0; 4 = trace internal.

Name	Type	Default value	Allowed values	When set	Description
trace_gicv3_its	bool	false	true, false	Init time	Trace GICv3 ITS command execution.
trace_gicv3_reads	bool	false	true, false	Init time	Trace GICv3 memory mapped reads.
trace_gicv5_stream_comm	bool	false	true, false	Init time	Trace GICv5<->CPUIF stream commands.
trace_gpt	bool	true	true, false	Init time	Trace packets generated by the GPT.
trace_hacdbss	bool	true	true, false	Init time	Trace packets generated by the HACDBS.
trace_hdbss	bool	true	true, false	Init time	Trace packets generated by the HDBSS.
trace_instructions	bool	true	true, false	Init time	Trace instructions.
trace_loads_store_memtype	bool	false	true, false	Init time	Show memory type information for core loads and stores.
trace_loads_stores	bool	true	true, false	Init time	Trace loads and stores that are triggered by instructions. These might go into the memory subsystem, into a cache, or into a TCM. This considerably slows down the model.
trace_mask_s_regs	bool	false	true, false	Init time	Represent non-updated bytes as ---- in S-registers trace.
trace_memory	bool	false	true, false	Init time	Trace memory accesses just outside the core.
trace_mmu	bool	true	true, false	Init time	Trace mmu tablewalks and associated information.
trace_mpu_events	bool	false	true, false	Init time	Trace MPU events.
trace_spe	bool	true	true, false	Init time	Trace SPE data written to memory.
trace_tag_loads_stores	bool	true	true, false	Init time	Trace tag loads and stores that are triggered by MTE instructions. These might go into the memory subsystem, into a cache, or into a TCM. This considerably slows down the model.
trace_vfp	bool	true	true, false	Init time	Trace the VFP and Neon registers, including FPSCR and FPEXC.
unbuffered	bool	false	true, false	Init time	Trace events as they arrive and flush each fwrite. Prints IT even when IS should be printed.
updated-registers	bool	false	true, false	Init time	Trace the updated value of registers rather than the written value.
use_inst_end_for_inst_counter	bool	false	true, false	Init time	When using the instruction count as the timestamp, if true, increase the instruction count at INST_END instead of INST. When using the simulation time as the timestamp, this parameter has no effect.
use_instr_cnt_as_timestamp	bool	true	true, false	Init time	Use the instruction count as the timestamp instead of the simulation time.

## 4.18.2 TarmacTrace file format

This topic describes conventions used in the syntax definitions for each trace type.

In the syntax definition for each trace type:

**[X|Y]**

Indicates a choice between X and Y.

**{X}**

Indicates that X is optional or configuration-dependent.

This is the common address definition that is used in the trace command syntax:

```
<vaddr>{ : <paddr><pas> }
```

where:

**<vaddr>**

is the virtual address in hexadecimal format. See the note after this list.

**<paddr>**

is the physical address of the instruction in hexadecimal. See the note after this list. <paddr> is only present if it is different to <vaddr>.

**<pas>**

is the address space of the physical address. Either **\_NS** for Non-secure PAS, **\_RT** for Root PAS, **\_RL** for Realm PAS, or not present for Secure PAS.



**Note**

For 64-bit addresses, the value is written as either:

- 8 hex digits, if the value can be represented in 32 bits.
- 16 hex digits otherwise.

This is the virtual regime definition that is used in the trace command syntax:

```
0x<vbase>{ _NS } <el>{ vmid=<vmid> }{ , nG asid=<asid> }
```

where:

**0x<vbase>**

is the virtual address in hexadecimal format.

**\_NS**

if present, specifies that the address is Non-secure. If not present, the address is Secure.

**<el>**

is the translation regime that owns the mapping. One of:

- **EL1\_n**, meaning the Non-secure EL1&0 translation regime.

- EL2\_n
- EL1\_s
- EL3\_s

**<vmid>**

if present, is the VMID for Non-secure, non-hyp regimes.

**nG**

if present, specifies that the virtual regime is non-global.

**<asid>**

if present, is the ASID, for non-global regimes.

### 4.18.3 Tarmac Trace output example

This example output from the Tarmac Trace plug-in shows various types of trace output, including instruction, memory access, register, translation table walk, and TLB traces.

```
...
89 clk IT (89) 80000164 f9000820 O EL1h_n : STR x0,[x1,#0x10]
89 clk MW8 80002010:000080002010_NS 00000000_80000705
90 clk IT (90) 80000168 d28080a0 O EL1h_n : MOV x0,#0x405
90 clk R X0 00000000000000405
...
98 clk IT (98) 80000188 d5181000 O EL1h_n : MSR SCTLR_EL1,x0
98 clk R SCTLR_EL1 00000000:00001005
98 clk TTW ITLB LPAE 1:1 000080002010 0000000080000705 : BLOCK ATTRIDX=1 NS=0 AP=0
SH=3 AF=1 nG=0 16E=0 PXN=0 XN=0 ADDR=0x0000000080000000
98 clk TLB FILL cpu0.UTLB 1G 0x80000000_NS EL1_n vmid=0:0x008000000000_NS Normal
InnerShareable Inner=WriteBackWriteAllocate Outer=WriteBackWriteAllocate_xn=0 pxn=0
ContiguousHint=0 xs=0
...
```

This trace shows three instructions:

- The first instruction is an `STR`, which stores the 64-bit value from register `x0` to the address in `x1 + 10` byte offset:

```
89 clk IT (89) 80000164 f9000820 O EL1h_n : STR x0,[x1,#0x10]
89 clk MW8 80002010:000080002010_NS 00000000_80000705
```

In more detail:

- `IT` is a label that indicates the type of trace event described by the line. `IT` means instruction taken. To interpret the values in this line, see [Instruction trace](#). For example:
  - `89` means this is the 89th instruction.
  - `clk` means that the preceding number is an instruction count. If `ps` was displayed here instead, this would indicate the first value is a timestamp.
  - `0x80000164` is the address from which the instruction was fetched.
  - `0xf9000820` is the 32-bit opcode of the instruction.

- o indicates the CPU execution state, in this case AArch64.
  - EL1h\_n indicates the current Exception level and Security state.
  - The rest of the line following the colon is the assembly language representation of the instruction.
- mw8 indicates an 8-byte memory write. To interpret the values in this line, see [Processor memory access trace](#). For example:
  - 0x80002010 is the virtual address to which the data was written. The value after the colon is the corresponding physical address. In this example, they are the same. The \_ns suffix indicates that it is Non-secure memory.
  - 0x00000000\_80000705 is the value of the data written. Each group of 8 digits is separated using an underscore.
- The second instruction is a mov, which moves the value 0x405 into register x0:

```
90 clk IT (90) 80000168 d28080a0 O EL1h_n : MOV x0,#0x405
90 clk R X0 00000000000000405
```

- r indicates a register trace. To interpret the values in this line, see [Register trace](#). For example:
  - x0 is the name of the register being written to.
  - 0x405 is the new value of x0, which is the value that was moved by the mov instruction.
- The third instruction is an msr, which writes the value 0x1005 from register x0 to System register SCTLR\_EL1.

Writing to bit 0 of SCTLR\_EL1 enables the MMU, so that all subsequent memory accesses will be done through the MMU.

The next memory access following this instruction is an instruction fetch (not shown), so a Translation Table Walk (TTW) is required to find its address.

Following the TTW, the Translation Lookaside Buffer (TLB) is updated with the new entry that caches some of the values resulting from the TTW, for example region size, base address, cachability and sharability. This appears as a TLB trace:

```
98 clk IT (98) 80000188 d5181000 O EL1h_n : MSR SCTLR_EL1,x0
98 clk R SCTLR_EL1 00000000:00001005
98 clk TTW ITLB LPAAE 1:1 000080002010 0000000080000705 : BLOCK ATTRIDX=1 NS=0
AP=0 SH=3 AF=1 nG=0 16E=0 PXN=0 XN=0 ADDR=0x0000000080000000
98 clk TLB FILL cpu0.UTLB 1G 0x80000000 NS EL1_n vmid=0:0x0080000000 NS
Normal InnerShareable Inner=WriteBackWriteAllocate Outer=WriteBackWriteAllocate
xn=0 pxn=0 ContiguousHint=0 xs=0
```

- TTW indicates a translation table walk trace. To interpret the values in this line, see [Translation table walk trace](#). For example:
  - ITLB means instruction TLB.
  - LPAAE means Large Physical Address Extension (LPAAE)-format translation table entries.



- 1:1 means Walk stage 1, Walk level 1.
- 0x000080002010 is the page base address and the page attributes.
- 0x0000000080000705 is the raw translation table entry.
- Following the colon is the parsed result. In this case, the LPAE region descriptor.
- TLB indicates a TLB trace. To interpret the values in this line, see [TLB trace](#). For example:
  - FILL means a TLB fill operation.
  - cpu0.UTLB means the operation is taking place on a Unified TLB, which is shared for I-side and D-side accesses.
  - 1G means the TLB entry is for a 1GB page.
  - 80000000\_ns means the entry has a page base address of 0x80000000 and is Non-Secure.
  - EL1\_n means the entry is for EL1.
  - vmid=0:0x008000000000\_ns means the entry is tagged with a specific VMID.
  - Normal InnerShareable means the entry is tagged as Normal Inner-Sharable.
  - Inner and outer are the inner and outer cache attributes for this entry.
  - xn=0 means the entry is tagged NOT Execute Never.
  - pxn=0 means the entry is tagged NOT Privileged Execute Never.
  - ContiguousHint=0 means the entry is not tagged as part of a set of contiguous entries that can be cached as one entry.
  - xs=0 means the page for this TLB entry is NOT XS, indicating either lack of support for FEAT\_XS in the core, or it is non-XS memory.

#### 4.18.4 Instruction trace

If enabled, this trace source generates one record for every instruction started.

The records (lines) of the instruction trace have this syntax:

```
<time> <scale> <cpu> [IT|IS] (<inst_id>) <addr> <opcode> [A|T|X|O] <mode>_<security> :
<disasm>
```

##### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

##### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

##### <cpu>

Processor, or other component, that gave the instruction.

**[IT|IS]****IT**

Instruction passed the condition code (taken).

**IS**

Instruction failed the condition code (skipped).

**<inst\_id>**

Tick count of this processor, which is equivalent to the number of instructions that are executed, except for certain instructions like `WFI/WFE` (decimal value).

**<addr>**

Fetch source address for this instruction. Formatted according to the common address definition, see [TarmacTrace file format](#).

**<opcode>**

16-bit or 32-bit hexadecimal opcode of the instruction.

**[A|T|X|O]**

Instruction set:

**A**

A32

**T**

T32

**X**

T32EE

**O**

A64

**<mode>**

Processor execution mode.

AArch32 modes are `svc`, `irq`, `fiq`, `usr`, `mon`, `sys`, `abt`, `und`, `hyp`.

AArch64 modes are `EL3h`, `EL3t`, `EL2h`, `EL2t`, `EL1h`, `EL1t`, `EL0t`.

**<security>**

Processor security state (`s` or `ns`).

**<disasm>**

Disassembly of the instruction.

## 4.18.5 Program flow trace

If enabled, every executed branch instruction triggers this trace source, which is a more efficient way to reconstruct the program flow than by tracing every instruction.

Output syntax:

```
<time> <scale> {<cpu>} [FD|FI] (<inst_id>) <addr> <targ_addr> [A|T|X|O]
```

### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

### <cpu>

Processor, or other component, that gave the instruction.

### [FD|FI]

Program flow change by:

#### FD

A direct branch.

#### FI

An indirect branch.

### <inst\_id>

Tick count of this processor, which is equivalent to the number of instructions that are executed, except for certain instructions like `WFI/WFE` (decimal value).

### <addr>

Fetch source address for this instruction. Formatted according to the common address definition, see [TarmacTrace file format](#).

### <targ\_addr>

Address (virtual) at which the execution continues. Formatted according to the common address definition.

### [A|T|X|O]

Instruction set after the branch:

#### A

A32.

#### T

T32.

#### X

T32EE.

#### O

A64

## 4.18.6 Register trace

If enabled, this source traces all writes to the processor registers.

This trace source includes writes to core registers `R0` to `R14`, `X0` to `X30`, `CPSR`, and `SPSR`, VFP registers such as `S0` to `S31`, `D0` to `D31`, `FPCR`, and `FPEXC`, and writes to system registers including `CP14`, `CP15`, and `GIC`. Banked registers are traced separately using the mode as a suffix to the register name, for example `r13` (current register `R13`) and `r13_mon` (banked register `R13`).

Output syntax:

`<time> <scale> {<cpu>} R <register> <value>`

### **<time>**

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

### **<scale>**

Unit for `<time>`. `ps` means simulation time, `clk` means instruction count.

### **<cpu>**

Processor, or other component, that gave the instruction.

### **<register>**

Register name. Banked core registers can have a mode appended to them with a single underscore. Banked `CP14/CP15` registers have `_s` or `_ns` appended to indicate access of either the Secure or Non-secure banked register.



Note

In Arm®v8 and Arm®v9, when the register name is `cpsr`:

- In AArch64 state, `cpsr` is used to trace PSTATE changes. The bit format of `<value>` follows the `SPSR_ELx` AArch64 format.
- In AArch32 state, the bit format of `<value>` follows the `CPSR` format.

### **<value>**

Hexadecimal value that is written to the register (64 bits maximum).

If the SVE plug-in is loaded in the model, there are additional registers in the program view. The output examples below show how these registers are traced when the value changes. These data values can be very large.

```
8463 clk cpu0 IT (8439) 000282c0:0000152282c0_NS 053fc01f O EL1h_n : SEL
z31.B,p0,z0.B,z31.B
8463 clk cpu0 R z31 00000000_00000000_00000000_00000000
```

**R** indicates a register write. **z0** to **z31** are the vector registers. The written data are hexadecimal digits, which are separated by an underscore every 32 bits. The length of the written data varies with the configuration, depending on the vector length.

```
9756 clk cpu0 IT (9732) 01000074:000011000074_NS 2518e3e0 O EL1t_n : PTRUE p0.B,ALL
9756 clk cpu0 R p0 ffff
```

**R** indicates a register write. **p0** to **p15** are the predicate registers. The written data are hexadecimal digits. If they are long enough to require one, the digits are separated by an underscore every 32 bits. The length of the written data varies with the configuration, depending on the vector length. Predicate registers contain 1 bit per byte of vector register length.

### 4.18.7 Cache maintenance trace

If enabled, traces all cache maintenance operations that the processor initiates.

Output syntax:

```
<time> <scale> <cpu> CACHE MAINTENANCE <side> <operation> <scope> <data> {<pagesize>
<mentype>}
```

#### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

#### <scale>

Unit for `<time>`. A value of `ps` means simulation time, `clk` means instruction count.

#### <cpu>

Processor, or other component, that gave the instruction.

#### <side>

Data or instruction cache.

#### <operation>

Clean, invalidate, or both.

#### <scope>

By MVA or set/way, to Point of Coherency or Point of Unification, Inner Sharable or not.

#### <data>

Data that is associated with the operation. If the operation is by MVA, formatted according to the common address definition, see [TarmacTrace file format](#), otherwise use raw hexadecimal.

#### <pagesize>

If the operation is by MVA, this element is the size of the memory region that is described by the TLB entry that contains the MVA.

**<memtype>**

If the operation is by MVA, this element is the type of memory in the TLB entry that contains the MVA.

## 4.18.8 Cache content trace

Traces the movement of data into and out of the cache.

Output syntax:

```
<time> <scale> <cpu> CACHE <id> LINE <line> <operation> 0x<paddr><ns>
```

**<time>**

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

**<scale>**

Unit for <time>. A value of `ps` means simulation time, `clk` means instruction count.

**<cpu>**

Processor, or other component, that gave the instruction.

**<id>**

Level and side, or system identifier, of the cache.

**<line>**

Identifier of this line uniquely within this cache, expressed in hexadecimal.

**<operation>**

Notification for this cache line. One of the following options:

**ALLOC**

(Processor caches) Line contains new read data.

**INVAL**

(Processor caches) Line contains no data.

**DIRTY**

(Processor caches) Line contains new write data.

**CLEAN**

(Processor caches) Write data is written back, still valid for reads.

**FILL**

(System caches) Line is filled.

**EVICT**

(System caches) Line is evicted due to space pressure.

**CLEAN**

(System caches) Line is cleaned due to maintenance operation.

**INVAL**

(System caches) Line is invalidated due to maintenance operation.

**<paddr>**

Cache line physical address in hexadecimal.

**<ns>**

Cacheline security. Blank for Secure regime, or `_ns` for Non-secure regime.

## 4.18.9 Translation table walk trace

If enabled, this source traces all translation table walks initiated by the processor.

Output syntax:

```
<time> <scale> <cpu> [TTW|TTU] <side> <format> <stage>:<level> <address> <data> :  
<result>
```

**<time>**

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

**<scale>**

Unit for `<time>`. `ps` means simulation time, `clk` means instruction count.

**<cpu>**

Processor, or other component, that gave the instruction.

**[TTW|TTU]**

Translation table walks (reads) or translation table update (writes).

**<side>**

Data or instruction TLB.

**<format>**

VMSA or LPAE format translation table entries.

**<stage>**

Walk stage, within the range 1-2.

**<level>**

Walk level, within the range 1-3.

**<address>**

Physical address of lookup in hexadecimal.

**<data>**

Raw translation table entry in hexadecimal.

**<result>**

Parsed result. One of the following options:

**ABORTED**

The memory access caused a synchronous abort and no data was returned.

**FAULT**

The data that was returned is not valid for this stage and level.

**RESERVED**

The data that was returned is not valid for this stage and level.

**TABLE** {<attr>=<value>}+

Pointer to the next level of lookup, in LPAAE format.

**BLOCK** {<attr>=<value>}

LPAAE region descriptor.

**SUPERSECTION** {<attr>=<value>}

VMSA region descriptor.

**SECTION** {<attr>=<value>}

VMSA region descriptor.

**PAGETABLE** {<attr>=<value>}

Pointer to the next level of lookup, in VMSA format.

**LARGEPAGE** {<attr>=<value>}

VMSA region descriptor.

**SMALLPAGE** {<attr>=<value>}

VMSA region descriptor.

## 4.18.10 Granule protection table walk trace

If enabled, this source traces Granule Protection Table (GPT) walks. This event is triggered by GPT lookups.

Output syntax:

```
<time> <scale> <cpu> GPTW [ISIDE|DSIDE] L<level> <address> <descaddr> : <data>
<result>
```

**<time>**

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

**<scale>**

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

**<cpu>**

Processor, or other component, that gave the instruction.

**[DSIDE|ISIDE]**

Data or instruction TLB.



**<level>**

GPT fetch level, either 0 or 1.

**<address>**

Physical address of the lookup.

**<descaddr>**

Physical address of the GPT descriptor.

**<data>**

GPT data that was read.

**<result>**

Parsed result. One of the following values:

**BLOCK TYPE=0x01 GPI=<gpi> PGS=<pgs>**

GPT descriptor is a Block descriptor. <gpi> is the GPI value of the fetched GPT entry. <pgs> is the physical granule size.

**CONTIGUOUS TYPE=0x01 GPI=<gpi> CRS=<crs>**

GPT descriptor is a Contiguous descriptor. <gpi> is the GPI value of the fetched GPT entry. <crs> is the contiguous region size.

**TABLE TYPE=0x03 ADDR=<addr>**

GPT descriptor is a Table descriptor. <addr> is the next-level table address in hexadecimal.

**GRANULE TYPE=0x0f GPI=<gpi> PGS=<pgs>**

GPT descriptor is a Granule descriptor. <gpi> is the GPI value of the fetched GPT entry. <pgs> is the physical granule size.

**INVALID**

GPT entry is invalid.

## 4.18.11 TLB trace

If enabled, this source traces TLB entries that are filled and evicted by the processor.

Output syntax:

```
<time> <scale> <cpu> [TLB|WALKCACHE] FILL <id> <size> <virtualregime>:<paddr>
{<mentype>} {<attr>=<value>}+
```

or

```
<time> <scale> <cpu> [TLB|WALKCACHE] EVICT <id> <size> <virtualregime>
```

**<time>**

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

**<scale>**

Unit for <time>. *ps* means simulation time, *clk* means instruction count.

**<cpu>**

Processor, or other component, that gave the instruction.

**<id>**

Identifies which TLB or walk cache to trace.

**<size>**

Size of the region being mapped.

**<virtualregime>**

Virtual address and regime of the region being mapped, formatted according to the common virtual regime definition.

**<paddr>**

Physical base address of mapped region, formatted according to the common address definition, see [TarmacTrace file format](#).

**<memtype>**

For TLB entries, the memory type of the result. One of the following options:

**Device- [G|nG] [R|nR] [E|nE] {(<alias>) }**

Device memory, where:

**[G|nG]**

Gathering or nongathering.

**[R|nR]**

Reordering or nonreordering.

**[E|nE]**

Early write acknowledgement or not.

**<alias>**

Device-nGnRnE was previously known as StronglyOrdered.

**Normal [NonShareable|Shareable] Inner=<cachetype> Outer=<cachetype>**

Normal memory, where:

**[NonShareable|Shareable]**

Shareability

**<cachetype>**

[NonCacheable|WriteBack|WriteThrough] {NonReadAllocate} {Non}  
{WriteAllocate}

**[NonCacheable|WriteBack|WriteThrough]**

Cacheability

**{NonReadAllocate}**

For cacheable memory, Read allocate hint. (Read allocate is assumed if not specified.)

**{Non}{WriteAllocate}**  
For cacheable memory, Write allocate hint.

4.18.12 Event trace

If enabled, traces exceptions, interrupts, and exception returns. In AArch64, it also traces changes to the SPSel and to the current exception level, by generating a CoreEvent\_ModeChange.

Output syntax:

<time> <scale> {<cpu>} E <value> {<mode>} {<value1>} <number> <desc>

**<time>**  
Either the simulation time timestamp or the instruction count, depending on the value of the TRACE.TarmacTrace.use\_instr\_cnt\_as\_timestamp parameter.

**<scale>**  
Unit for <time>. ps means simulation time, clk means instruction count.

**<cpu>**  
Processor, or other component, that gave the instruction.

**<value>**  
A value that is associated with the event, formatted according to the common address definition, see [TarmacTrace file format](#).

**<mode>**  
For mode change events only, the new mode being entered.

**<value1>**  
Where available, the hexadecimal representation of a second value that is associated with the event.

**<number>**  
Event number.

**<desc>**  
Event name.

In the following table, the CoreEvent\_CURRENT\_\* and CoreEvent\_LOWER\_\* events cover all the ways in which exception entry can happen in AArch64 state. For example, CoreEvent\_CURRENT\_SPx\_SYNC corresponds to a synchronous exception taken from Current Exception level with SP\_ELx, x>0.

CoreEvent\_LOWER\_64\_IRQ corresponds to an IRQ or vIRQ taken from Lower Exception level, where the implemented level immediately lower than the target level is using AArch64.

Table 4-17: Supported values for value, number, and desc

Number	Event description	Value
0x00000001	CoreEvent_Reset	-

Number	Event description	Value
0x00000002	CoreEvent_UndefinedInstr	-
0x00000003	CoreEvent_SWI	SWI number
0x00000004	CoreEvent_PrefetchAbort	-
0x00000005	CoreEvent_DataAbort	-
0x00000007	CoreEvent_IRQ	-
0x00000008	CoreEvent_FIQ	-
0x0000000E	CoreEvent_ImpDataAbort	-
0x00000019	CoreEvent_ModeChange	New mode
0x00000080	CoreEvent_CURRENT_SPO_SYNC	-
0x00000081	CoreEvent_CURRENT_SPO_IRQ	-
0x00000082	CoreEvent_CURRENT_SPO_FIQ	-
0x00000083	CoreEvent_CURRENT_SPO_ABORT	-
0x00000084	CoreEvent_CURRENT_SPx_SYNC	-
0x00000085	CoreEvent_CURRENT_SPx_IRQ	-
0x00000086	CoreEvent_CURRENT_SPx_FIQ	-
0x00000087	CoreEvent_CURRENT_SPx_ABORT	-
0x00000088	CoreEvent_LOWER_64_SYNC	-
0x00000089	CoreEvent_LOWER_64_IRQ	-
0x0000008A	CoreEvent_LOWER_64_FIQ	-
0x0000008B	CoreEvent_LOWER_64_ABORT	-
0x0000008C	CoreEvent_LOWER_32_SYNC	-
0x0000008D	CoreEvent_LOWER_32_IRQ	-
0x0000008E	CoreEvent_LOWER_32_FIQ	-
0x0000008F	CoreEvent_LOWER_32_ABORT	-

### 4.18.13 Processor memory access trace

If enabled, this source traces processor data accesses.

Output syntax:

```
<time> <scale> {<cpu>} M<rw><sz><attrib> <addr><data>
```

**<time>**

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

**<scale>**

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

**<cpu>**

Processor, or other component, that gave the instruction.

**<rw>****R**

Read access.

**W**

Write access.

**<sz>**

Size of the data transfer in bytes (1, 2, 4, 8).

**<attrib>**

Optional access attribute:

**X**

Exclusive access.

**T**

Translated (unprivileged) access.

**L**

Locked access (SWP, SWPB instructions).

**\_CAS<suffix>**

Compare and swap operation, where <suffix> is either *c* or *d*. *\_CASc* shows the value to compare and *\_CASd* shows the value that will be written to memory if the comparison matches.

**Note**

The value that is stored in memory as a result of a compare and swap operation is shown by an *MU* trace source.

**<addr>**

Virtual address that is used to access memory. Formatted according to the common address definition, see [TarmacTrace file format](#).

**<data>**

Hexadecimal value of data transferred. The data padding is according to the size of the transfer. Data of 64 bits or more contains an underscore (*\_*) separator every eight characters (32 bits).

## 4.18.14 Processor memory update trace

If enabled, this source traces memory update accesses caused by atomic operations.

Output syntax:

```
<time> <scale> {<cpu>} MU<sz>_<atomic_op> <addr> <data>
```

**<time>**

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

**<scale>**

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

**<cpu>**

Processor, or other component, that gave the instruction.

**<sz>**

Size of the data transfer in bytes (1, 2, 4, 8, 16).

**<atomic\_op>**

Atomic operation performed on this memory address:

**ADD**

Atomic add operation.

**BIC**

Atomic bit clear operation.

**CASc**

Atomic compare and swap operation.

**EOR**

Atomic exclusive or operation.

**ORR**

Atomic bit set operation.

**SMAX**

Atomic signed max operation.

**SMIN**

Atomic signed min operation.

**SWP**

Atomic swap operation.

**UMAX**

Atomic unsigned max operation.

**UMIN**

Atomic unsigned min operation.

**<addr>**

Physical address that is used to access memory. Formatted according to the common address definition, see [TarmacTrace file format](#).

**<data>**

Hexadecimal value of data that is stored in memory as a result of the atomic operation. Data of 64 bits or more contains an underscore ( `_` ) separator every eight characters (32 bits).

## 4.18.15 Memory bus trace

If enabled, this source traces transactions that are initiated through the memory bus master port of the processor. These accesses use physical addresses.

Output syntax:

```
<time> <scale> {<cpu>} B<rw><sz><fd><lk><p><s> I<wrcbs> O<wrcbs> <master_id> <addr>
<data>
```

### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

### <cpu>

Processor, or other component, that gave the instruction.

### <rw>

**R**

Read access.

**W**

Write access.

### <sz>

Size of the data transfer in bytes.

### <fd>

**I**

Opcode fetch.

**D**

Data load/store or an MMU access.

### <lk>

**L**

Locked access.

**X**

Exclusive access.

**\_**, **underscore**

Normal access.

**<p>****P**

Privileged access.

**\_, underscore**

Normal access.

**<s>****S**

Secure access.

**N**

Non-secure access.

**I<wrcbs>**

Inner cache attributes. See o&lt;wrcbs&gt;.

**O<wrcbs>**

Outer cache attributes:

**<w>****W**

Allocate on write.

**\_, underscore**

No allocate on write.

**<r>****R**

Allocate on read.

**\_, underscore**

No allocate on read.

**<c>****C**

Cacheable access.

**\_, underscore**

Non-cacheable access.

**<b>****B**

Bufferable access.

**\_, underscore**

Non-bufferable access.



**<s>**

**s**

Shareability access.

**\_, underscore**

Non-shareability access.

**<master\_id>**

Master ID of the transaction.

**<addr>**

Physical address that is used to access memory, in hexadecimal format.

**<data>**

Hexadecimal value of data transferred. The data padding is according to the size of the transfer. Byte ordering is from lowest to highest byte. This ordering means that for accesses in little endian mode, the data occurs mirrored compared to the register/memory access records.

## 4.19 ToggleMTIPlugin

ToggleMTIPlugin is an MTI plug-in that can be used to limit the generation of trace by another plug-in to specific areas of interest.

Generating trace output throughout a simulation session can reduce simulation speed and result in very large trace files. ToggleMTIPlugin helps to avoid these problems by enabling you to toggle trace generation during the simulation. Toggling trace means that if trace is on, it is turned off, and vice versa.

ToggleMTIPlugin can be used with the following plug-ins:

- ASTFplugin
- GenericTrace
- TarmacTrace

### 4.19.1 ToggleMTIPlugin - parameters

This section describes the parameters for the ToggleMTIPlugin plug-in.

Each parameter is prefixed with `TRACE.ToggleMTIPlugin`, for example:

```
TRACE.ToggleMTIPlugin.diagnostics
```

**Table 4-18: ToggleMTIPlugin parameters**

Name	Type	Default value	Allowed values	When set	Description
diagnostics	bool	false	true, false	Init time	Print diagnostics.
disable_mti_from_start	bool	false	true, false	Init time	Enable or disable MTI callbacks from start of simulation.
disable_mti_runtime	bool	false	true, false	Runtime	Enable or disable MTI callbacks at runtime.
hlt_imm16	int	0xf000	0x0 - 0xffff	Init time	16-bit integer used in HLT instruction meant to be used by this plugin.
use_hlt	bool	true	true, false	Init time	If true, use HLT #imm16 instruction to toggle MTI behavior.

### 4.19.2 How to use ToggleMTIPlugin

As with other plug-ins, load ToggleMTIPlugin using the `--plugin` command-line option when launching the model.



**Note**

- When loading ToggleMTIPlugin and any other trace plug-ins using the `--plugin` option, ToggleMTIPlugin must be the last plug-in to be specified on the command line.
- We recommend you disable trace generation from the start of the simulation, using the plug-in parameter `disable_mti_from_start=1`, then enable it when execution reaches the region of interest.

There are two alternative ways to use ToggleMTIPlugin. You cannot use both in the same simulation session. Use the `use_hlt` plug-in parameter to control which one to use:

- `use_hlt = 1`

To use this method, set the `hlt_imm16` plug-in parameter to an integer value. The application will use this value as the operand in `HLT` instructions to toggle MTI callbacks.

You must also set the following parameters on the core model that is running the application:

#### **enable\_trace\_special\_hlt\_imm16**

Set to true to enable the parameter `trace_special_hlt_imm16`.

#### **trace\_special\_hlt\_imm16**

Specifies the integer value that is used as the operand to `HLT` instructions to cause the usual `HLT` execution to be skipped. If the value matches the value specified in the `hlt_imm16` plug-in parameter, tracing is toggled.

- `use_hlt = 0`

To use this method, set the runtime plug-in parameter `disable_mti_runtime` during the simulation session to either true to disable tracing, or false to enable tracing. Changes to the `disable_mti_runtime` parameter are ignored unless `use_hlt` is zero.

To change `disable_mti_runtime` at runtime, use a debugger, for example Iris Monitor or use the `iris.debug` Python module. The example Python script `$PVLIB_HOME/examples/pyIris/inst_count_trace_control.py`, demonstrates how to do this.

## 5. Fast Models examples

The following top-level example directories are installed under `$PVLIB_HOME/examples/`.

**Table 5-1: Fast Models examples directories**

Directory name	Description
LISA	LISA+ source code and project files for FVPs.
LISAPlus	Example LISA+ components that show how to capture and generate MTI trace, remap PVBUS transactions, and handle burst transactions.
MTI	MTI plug-in examples that show how to extract and use trace information from models. These examples are also available as pre-built libraries under <code>\$PVLIB_HOME/plugins/</code> : <ul style="list-style-type: none"> <li>GenericCounter</li> <li>GenericTrace</li> <li>ListTraceSources</li> </ul>
SystemCExport	<ul style="list-style-type: none"> <li>Source code and makefiles for EVS platform examples and SVPs</li> <li>LISA+ source for bridges and EVS components</li> <li>Header files required for exporting LISA+ protocols to SystemC</li> </ul>



On Microsoft Windows, the Fast Models installer creates a copy of the examples in `%USERPROFILE%\ARM\FastModelsPortfolio_%FM-VERSION%\examples\`. This copy allows you to save configuration changes to the examples without needing Administrator permissions.

### 5.1 LISA examples

LISA+ source and project files for FVPs.



The LISA platform examples are Integrated SIMulators (ISIMs). For more information about building and running them, see [Build and run an FVP example](#).

The following LISA examples are provided:

**Table 5-2: LISA examples**

Example	Description
BusComponents	Example LISA+ components that demonstrate different ways of using the PVBUS interface.
Common	FVP-specific LISA+ components that are common to different types of FVPs.
CSS	Source and project files for Reference Design FVPs. These FVPs model compute subsystems (CSS) that target specific market segments. Reference software stacks are available for them, see <a href="#">Arm Ecosystem FVPs</a> for more information.

Example	Description
FVP_Base	Source and project files for Base Platform FVP examples. For information about the Base Platform, see <a href="#">Base Platform</a> .
FVP_BaseR	Source and project files for BaseR Platform FVP examples.
FVP_Base_RevC	Source and project files for Base Platform RevC FVP examples. For information about the Base Platform RevC, see <a href="#">Base Platform RevC</a> .
FVP_Coproc_Demo	Example implementation of the <code>Coprocessor</code> interface. Registers the coprocessor with a ARMCortexM33CT or ARMAEMv8MCT model. For more information, see <a href="#">CoprocBusProtocol protocol</a> .
FVP_MPS2	Source and project files for MPS2-based example platforms. For information about the MPS2 platforms, see <a href="#">MPS2</a> .
FVP_MPS3	Source and project files for MPS3-based example platforms that support the Arm®Corstone™ SSE-300 Example Subsystem. For more information, see <a href="#">Arm Corstone SSE-300 Example Subsystem Technical Reference Manual</a> .
FVP_VE	Source and project files for VE FVPs. For information about the VE platform, see <a href="#">Versatile Express</a> .
VP_PChannel	Shows how to create power controllers to control the power state of the cores and cluster, using the PChannel protocol. For information about PChannel, see <a href="#">PChannel protocol</a> .

## 5.2 Build and run an FVP example

The FVP examples are located under `$PVLIB_HOME/examples/LISA/`.

### Before you begin

Ensure the following:

- You have installed Fast Models and SystemC 2.3.4 and have set the environment variables as described in [Installation](#) in the Fast Models User Guide.
- You have set up a Fast Models license, either using FlexNet Licensing or [User-Based Licensing](#).
- You are using a supported Operating System and have set up a compatible toolchain, from those listed in [Requirements for Fast Models](#) in the Fast Models User Guide.

### About this task

You can build an FVP example in either of the following ways:

- From a terminal, using the `simgen` command.
- From System Canvas.

### Procedure

1. To build an FVP example from a terminal using `simgen`, enter the following commands:

```
cd $PVLIB_HOME/examples/LISA/FVP_Base/Build_Cortex-A55
simgen -b -p FVP_Base_Cortex-A55.sgproj --configuration Linux64-Release-GCC-9.3
```

where:

**-b**

means build the target.

**-p**

specifies the SimGen project file, with a `.sgproj` extension.

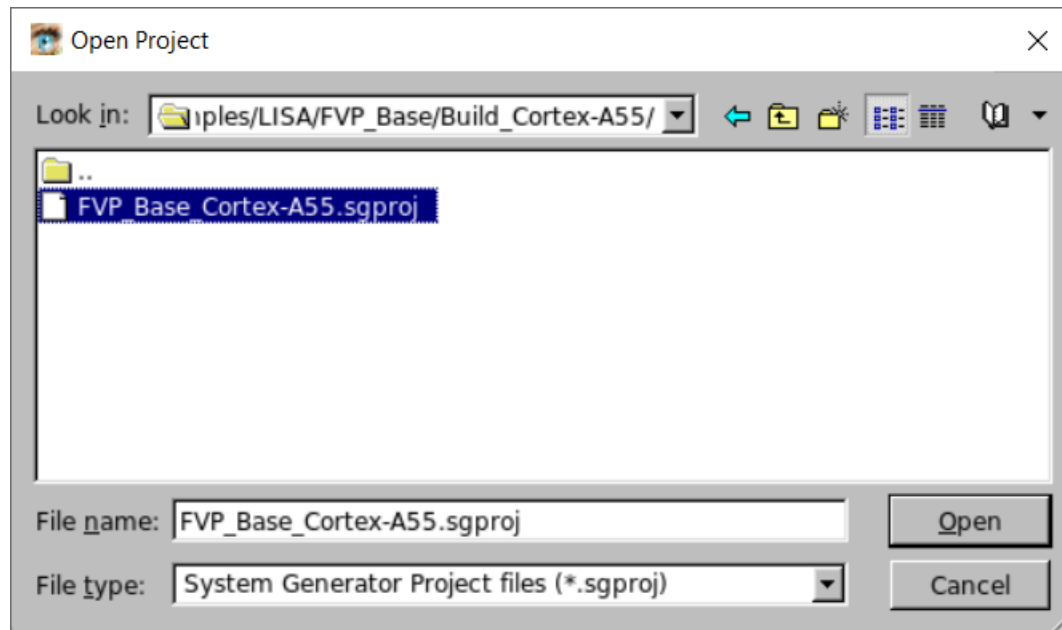
**--configuration**

specifies the build configuration name. In this example, we are performing a release build on Linux using GCC 9.3.

This command generates an executable named `isim_system` in `$PVLIB_HOME/examples/LISA/FVP_Base/Build_Cortex-A55/Linux64-Release-GCC-9.3/`.

2. To use System Canvas:
  - a. Start System Canvas by opening a terminal and typing `sgcanvas`.
  - b. In System Canvas, select **File > Load Project...** to load the `.sgproj` file for the example you want to build. This example uses `$PVLIB_HOME/examples/LISA/FVP_Base/Build_Cortex-A55/FVP_Base_Cortex-A55.sgproj`.

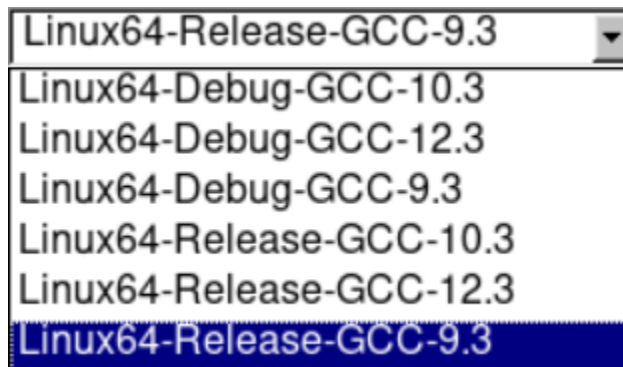
**Figure 5-1: System Canvas Open Project dialog box**



While the project is loading, you can ignore any errors like the following that might appear in the Output window:

```
Error (5104): undefined compile time parameter 'uart_base' in numeric expression
```

- c. Select the build configuration from the Active Project Configuration drop-down menu on the main toolbar:

**Figure 5-2: Select Active Project Configuration menu**

- d. Click **Build** to build the FVP executable.

If you changed the active project configuration, click **Yes** when prompted to save the modified project file. The output from the build process is shown in the output window at the bottom of System Canvas. If the build is successful, the last message displayed is **Model Build process completed successfully**.

- e. The generated executable is named `isim_system`. In this example, it is created in `$PVLIB_HOME/examples/LISA/FVP_Base/Build_Cortex-A55/Linux64-Release-GCC-9.3`.
3. You can run `isim_system` either from the terminal or from within System Canvas:
- To run `isim_system` from the terminal:
    - a. Navigate to the directory where it is located.
    - b. To see a full list of command-line options for `isim_system`, run it with the `--help` option:

```
./isim_system --help
```

- c. The following example command line shows how to load an application on `isim_system`:

```
./isim_system -a FVP_Base_Cortex_A55.cluster0.*=$PVLIB_HOME/images/image.axf -C bp.secure_memory=0
```

where:

**-a**

is the name of the application to load. Optionally it also specifies which core instances to run it on, in this case all cores in cluster0.

**-C**

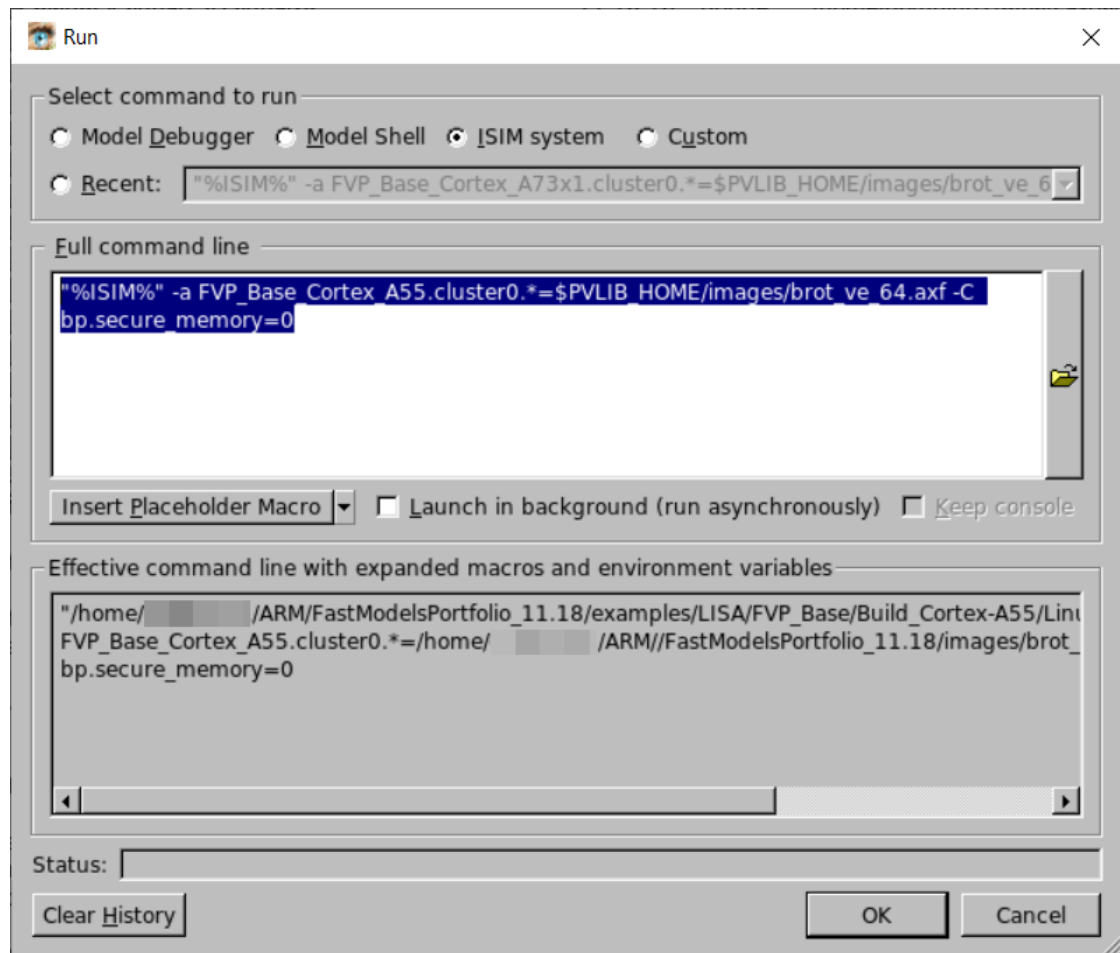
is a configuration parameter. To see a full list of the available parameters, run the model with the `--list-params` option. To specify multiple parameters, it can be more convenient to place them in a text file, each parameter on a new line, and pass them to the model when starting it using `--config-file <filename>`.

- d. The ISIM starts running, displaying the output in a CLCD window.
- e. To exit the simulation, press **Ctrl+C**.

- To run `isim_system` in System Canvas:
  - a. Click Run.
  - b. In the Run dialog box, select the **ISIM system** checkbox, then enter the following command under **Full command line**:

```
"%ISIM%" -a FVP_Base_Cortex_A55.cluster0.*=$PVLIB_HOME/images/image.axf -C  
bp.secure_memory=0
```

**Figure 5-3: System Canvas Run dialog box**



- c. Click **OK**. The ISIM starts running, displaying the output in a CLCD window.
- d. To exit the simulation, click **Kill**.

### Next steps

See the [System Canvas tutorial](#) in the Fast Models Tools User Guide for information on modifying, rebuilding, and debugging the example.

### Related information

[System Generator](#)



[System Canvas](#)[Fast Models glossary](#)[LISA examples](#) on page 5224

## 5.3 LISAPlus examples

Source code and System Canvas project files for some example LISA+ components.

**Table 5-3: LISAPlus examples**

Example	Description
CapturingTraceFromLISA	Instantiates and uses an MTI plug-in from a LISA+ component. Uses the SimpleTrace plug-in as an example.
GeneratingTraceFromLISA	Generates MTI trace information from a LISA+ component.
PVBusBursts	Uses a PVBusMaster to generate burst read transactions, which are handled by a PVBusSlave.
RemappingWithPVBusMapper	Uses the PVBusMapper component to remap transactions based on their attributes.

### Related information

[PVBusMapper](#) on page 201[PVBusMaster](#) on page 202[PVBusSlave](#) on page 206

## 5.4 MTI examples

Example MTI plug-ins that show how to use MTI to extract and use trace information from models.

The following MTI examples are provided:

**Table 5-4: MTI examples**

Example	Description
CallTrace	Displays a function call sequence by tracing the PC field of INST trace sources, then compares the output with values in a symbol table. See the <a href="#">readme</a> for more information.
CountingCacheStats	Registers counters for cache-related trace sources, for example CACHE_READ_HIT. Prints the cache stats before terminating.
DCCTrace	Prints the value of DBGDTRXX_EL0 when data is written. Updates the TxFull bit in MDSCR to indicate the data was read. See the <a href="#">readme</a> for more information.
GenericCounter	Registers a counter for trace sources. Prints the counter value for each INST trace source before terminating.  This example is also available as a pre-built library, see <a href="#">GenericCounter</a> .
GenericTrace	A flexible plug-in that traces one or more trace sources specified by the user. Prints the trace to a text file or to stdout.  This example is also available as a pre-built library, see <a href="#">GenericTrace</a> .

Example	Description
ITMtrace	Captures instrumentation trace macrocell (ITM) packets, which enables you to use ITM with a Cortex®-M class model.  For more information about this plug-in, see <a href="#">Trace Cortex-M software with the Instrumentation Trace Macrocell (ITM)</a> on Arm Community.
ListTraceSources	Displays either the trace sources provided by all trace components in the model, or just the trace components, to a text file or to stdout, without running the simulation. For more information, see <code>readme.txt</code> .  This example is also available as a pre-built library, see <a href="#">ListTraceSources</a> .
RunTimeParameterTest	Uses MTI to set runtime parameters.
SimpleTrace	Simple trace plug-in that prints a trace of the PC.
SoftwareTrigger	Traces SEMIHOSTING_PRECALL trace events, intercepts semihosting calls, and prints out register information. For more information, see the <code>readme</code> .
TraceOnBreak	Similar to the SimpleTrace example, but prints the PC value only when a breakpoint is hit.

## Related information

[Fast Models Model Trace Interface Reference Manual](#)

## 5.5 SystemCExport examples

Components and platform models that are created by exporting LISA+ components or platforms to SystemC. Also, bridge components for converting transactions between LISA+ protocols and SystemC.

**Table 5-5: SystemCExport examples**

Directory	Description
Bridges	LISA+ source for <a href="#">bridge components</a> .
Common	Source files and makefile rules that are common to the EVS and SVP examples.
Common/ Protocols	Header files that are required for the export of LISA+ protocols to SystemC.
EVS_Components	LISA+ files and project files for EVS (Exported Virtual Subsystem) components. These are LISA+ components with a SystemC wrapper and bridges that allow them to be used in a SystemC simulation.
EVS_Platforms	LISA+ source and makefiles for EVS platform examples.  An EVS platform is a LISA+ platform that has been exported as a SystemC object to allow it to be integrated into a SystemC simulation.  The EVS platform examples are minimal platforms that are designed for a specific use case, for example running the Dhrystone benchmark application or booting Linux.  The Dhrystone image <code>dhrystone_v8.axf</code> is provided in the Fast Models Third Party IP package. It is loaded from <code>\$PVLIB_HOME/images/</code> .  For more information about building an EVS platform, see <a href="#">Build and run an EVS platform example</a> .

Directory	Description
SVP_Platforms	<p>SVPs (SystemC Virtual Platforms) are platform models in which each component or subsystem has been individually exported to SystemC using the Fast Models Multiple Instantiation (MI) feature. For more information, see <a href="#">SVP build target</a> in the Fast Models User Guide.</p> <p>SVP platforms can be modified by replacing EVS components with other Fast Models EVSs, or with native SystemC components.</p> <p>For more information about building an SVP, see <a href="#">Build and run an SVP example</a>.</p>

## 5.6 Build and run an EVS platform example

The EVS platform examples are located under `$PVLIB_HOME/examples/SystemCExport/EVS_Platforms/`.

### Before you begin

Ensure the following:

- You have installed Fast Models and SystemC 2.3.4 and have set the environment variables as described in [Installation](#) in the Fast Models User Guide.
- You have set up a Fast Models license, either using FlexNet Licensing or [User-Based Licensing](#).
- You are using a supported Operating System and have set up a compatible toolchain, from those listed in [Requirements for Fast Models](#) in the Fast Models User Guide.

### About this task

Follow these instructions to build and run one of the EVS platform examples. This tutorial uses the `EVS_Dhrystone_Cortex-A75x1` example, which is a minimal platform that is designed to run the Dhrystone benchmark application.

### Procedure

1. These examples are built using a Makefile. Open a terminal and navigate to the directory containing the example, and run `make`, specifying the build configuration, for example:

```
cd $PVLIB_HOME/examples/SystemCExport/EVS_Platforms/EVS_Dhrystone/Build_Cortex-A75x1
make rel_gcc93_64
```

This command creates the target executable `EVS_Dhrystone_Cortex-A75x1.x` and copies into the current directory the shared object files that are required to run it.

2. To see a full list of command-line options for the EVS platform, run it with the `--help` option:

```
./EVS_Dhrystone_Cortex-A75x1.x --help
```

3. The Dhrystone image is available in the Third Party Add-ons for Fast Models package. Download this package from [Product Download Hub](#).
4. Install the Third Party Add-ons for Fast Models package to the location of your existing Fast Models installation. The package installs some example images, including `dhrystone_v8.axf` into `$PVLIB_HOME/images/`.

- From the terminal, enter the following command to launch the platform and load the Dhrystone image, specifying 10000000 as the number of runs through the benchmark:

```
echo 10000000 | ./EVS_Dhrystone_Cortex-A75x1.x -a $PVLIB_HOME/images/dhrystone_v8.axf
```

- The simulation runs, printing output to the terminal.

### Next steps

See the [System Canvas tutorial](#) in the Fast Models Tools User Guide for information on modifying, rebuilding, and debugging the example.

## 5.7 Build and run an SVP example

The SystemC Virtual Platform (SVP) examples are located under `$PVLIB_HOME/examples/SystemCExport/SVP_Platforms/`.

### Before you begin

Ensure the following:

- You have installed Fast Models and SystemC 2.3.4 and have set the environment variables as described in [Installation](#) in the Fast Models User Guide.
- You have set up a Fast Models license, either using FlexNet Licensing or [User-Based Licensing](#).
- You are using a supported Operating System and have set up a compatible toolchain from those listed in [Requirements for Fast Models](#) in the Fast Models User Guide.

### About this task

Follow these instructions to build and run one of the SVP examples.

### Procedure

- These examples are built using a Makefile. Open a terminal, navigate to the directory containing the example, and run `make`, specifying the build configuration, for example:

```
cd $PVLIB_HOME/examples/SystemCExport/SVP_Platforms/SVP_Base/Build_Cortex-A57x1
make rel_gcc93_64
```

This command creates the target executable `svp_Base_Cortex-A57x1.x` and copies into the current directory the shared object files that are required to run the platform.

- To see a full list of command-line options for the platform, run it with the `--help` option:

```
./SVP_Base_Cortex-A57x1.x --help
```

- Optionally, some example images are available in the Third Party Add-ons for Fast Models package. Download this package from [Product Download Hub](#).
- Install the package to the location of your existing Fast Models installation. The package installs the images into `$PVLIB_HOME/images/`.

5. The following example command-line launches the platform, using the `-a` option to load one of the images:

```
./SVP_Base_Cortex-A57x1.x -a $PVLIB_HOME/images/brot_ve_64.axf
```

6. The simulation starts running, displaying the output in a CLCD window.
7. To exit the simulation, press **Ctrl+C**.

### Next steps

See the [System Canvas tutorial](#) in the Fast Models Tools User Guide for information on modifying, rebuilding, and debugging the example.

# Proprietary Notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication, or disclosure of this document complies fully with any relevant

export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

# Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in Arm documents.

## Product status

All products and services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

### Product completeness status

The information in this document is Final, that is for a developed product.

## Revision history

These sections can help you understand how the document has changed over time.

### Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

#### Document history

Issue	Date	Confidentiality	Change
1129-00	16 May 2025	Non-Confidential	Update for v11.29
1128-00	19 February 2025	Non-Confidential	Update for v11.28
1127-00	16 September 2024	Non-Confidential	Update for v11.27
1126-00	19 June 2024	Non-Confidential	Update for v11.26
1125-00	13 March 2024	Non-Confidential	Update for v11.25.
1124-00	6 December 2023	Non-Confidential	Update for v11.24.



Issue	Date	Confidentiality	Change
1123-00	13 September 2023	Non-Confidential	Update for v11.23.
1122-00	14 June 2023	Non-Confidential	Update for v11.22.
1121-00	22 March 2023	Non-Confidential	Update for v11.21.
1120-00	7 December 2022	Non-Confidential	Update for v11.20.
1119-00	14 September 2022	Non-Confidential	Update for v11.19.
1118-00	15 June 2022	Non-Confidential	Update for v11.18.
1117-00	16 February 2022	Non-Confidential	Update for v11.17.
1116-00	6 October 2021	Non-Confidential	Update for v11.16.
1115-00	29 June 2021	Non-Confidential	Update for v11.15.
1114-01	30 March 2021	Non-Confidential	Document update 1 for v11.14.
1114-00	17 March 2021	Non-Confidential	Update for v11.14.
1113-00	9 December 2020	Non-Confidential	Update for v11.13.
1112-00	22 September 2020	Non-Confidential	Update for v11.12.
1111-00	9 June 2020	Non-Confidential	Update for v11.11.
1110-00	12 March 2020	Non-Confidential	Update for v11.10.
1109-00	28 November 2019	Non-Confidential	Update for v11.9.
1108-00	5 September 2019	Non-Confidential	Update for v11.8.

Issue	Date	Confidentiality	Change
1107-00	17 May 2019	Non-Confidential	Update for v11.7.
1106-01	25 March 2019	Non-Confidential	Update for v11.6.1.
1106-00	27 February 2019	Non-Confidential	Update for v11.6.
1105-00	23 November 2018	Non-Confidential	Update for v11.5.
1104-01	17 August 2018	Non-Confidential	Update for v11.4.2.
1104-00	22 June 2018	Non-Confidential	Update for v11.4.
1103-00	23 February 2018	Non-Confidential	Update for v11.3.
1102-00	17 November 2017	Non-Confidential	Update for v11.2.
1101-00	31 August 2017	Non-Confidential	Update for v11.1.
1100-00	31 May 2017	Non-Confidential	Update for v11.0. Document numbering scheme has changed.

## Change history

For information about the functional changes to Fast Models, see the [Fast Models Release Notes](#).

## Conventions

The following subsections describe conventions used in Arm documents.

### Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: [developer.arm.com/glossary](https://developer.arm.com/glossary).

### Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
<i>italic</i>	Citations.
<b>bold</b>	Interface elements, such as menu names.  Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments.  For example: <div>MRC p15, 0, &lt;Rd&gt;, &lt;CRn&gt;, &lt;CRm&gt;, &lt;Opcode_2&gt;</div>
SMALL CAPITALS	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, <b>IMPLEMENTATION DEFINED</b> , <b>IMPLEMENTATION SPECIFIC</b> , <b>UNKNOWN</b> , and <b>UNPREDICTABLE</b> .



We recommend the following. If you do not follow these recommendations your system might not work.



Your system requires the following. If you do not follow these requirements your system will not work.



You are at risk of causing permanent damage to your system or your equipment, or harming yourself.



This information is important and needs your attention.



A useful tip that might make it easier, better or faster to perform a task.



**Remember**

A reminder of something important that relates to the information you are reading.

---

# Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Arm documents are available on [developer.arm.com/documentation](https://developer.arm.com/documentation).

Confidential documents are only available to licensees, when logged in. Each document link in the tables below provides direct access to the online version of the document.

Arm product resources	Document ID	Confidentiality
<a href="#">A-Profile Architecture</a>	–	Non-Confidential
<a href="#">AMBA-PV Extensions to TLM User Guide</a>	100962	Non-Confidential
<a href="#">AN521 Example SSE-200 Subsystem for MPS2+ Application Note</a>	DAI 0521	Non-Confidential
<a href="#">Arm® Architecture Models</a>	–	Non-Confidential
<a href="#">Arm® Corstone™ SSE-300 Example Subsystem Technical Reference Manual</a>	101773	Non-Confidential
<a href="#">Arm® Cortex®-M23 Armv8-M IoT Kit User Guide</a>	ECM 0635473	Non-Confidential
<a href="#">Arm® Cortex®-M33 processor Armv8-M IoT Kit FVP User Guide</a>	ECM 0601256	Non-Confidential
<a href="#">Arm® Cortex®-M7 SMM on V2M-MPS2 Application Note 400</a>	DAI 0400	Non-Confidential
<a href="#">Arm® Development Studio Getting Started Guide</a>	101469	Non-Confidential
<a href="#">Arm® DynamIQ Shared Unit Technical Reference Manual</a>	100453	Non-Confidential
<a href="#">Arm® Socrates™</a>	–	Non-Confidential
<a href="#">Fast Models FVPs in Arm Development Studio Reference Guide</a>	110379	Non-Confidential
<a href="#">Fast Models Model Trace Interface Reference Manual</a>	DUI 0819	Non-Confidential
<a href="#">Fast Models Tools User Guide</a>	109415	Non-Confidential
<a href="#">Fast Models User Guide</a>	100965	Non-Confidential
<a href="#">Fixed Virtual Platforms</a>	–	Non-Confidential
<a href="#">How to generate ASTF traces of workloads running on Fast Models</a>	109193	Non-Confidential
<a href="#">Introduction to SVE2</a>	102340	Non-Confidential
<a href="#">LISA+ Language for Fast Models Reference Guide</a>	101092	Non-Confidential
<a href="#">Motherboard Express µATX V2M-P1 Technical Reference Manual</a>	DUI 0447	Non-Confidential
<a href="#">Product Download Hub</a>	–	Non-Confidential
<a href="#">Workload Trace Generation Best Practices</a>	107983	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
<a href="#">AMBA® Low Power Interface Specification</a>	IHI 0068	Non-Confidential
<a href="#">Armv7-M Architecture Reference Manual</a>	DDI 0403	Non-Confidential
<a href="#">Arm® Architecture Reference Manual Armv7-A and Armv7-R edition</a>	DDI 0406	Non-Confidential
<a href="#">Arm® Architecture Reference Manual Supplement, The Scalable Vector Extension</a>	DDI 0584	Non-Confidential
<a href="#">Arm® Architecture Reference Manual for A-profile architecture</a>	DDI 0487	Non-Confidential
<a href="#">Arm® Generic Interrupt Controller Architecture version 2.0 - Architecture Specification</a>	IHI 0048	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
<a href="#">Arm® Power Policy Unit Architecture Specification</a>	DEN 0051	Non-Confidential
<a href="#">Arm® Realm Management Extension (RME) System Architecture</a>	DEN 0129	Non-Confidential

Non-Arm resources	Document ID	Organization
<a href="#">Accellera Systems Initiative</a>	–	<a href="https://www.accellera.org">https://www.accellera.org</a>
<a href="#">GitHub</a>	–	Github
<a href="#">Intel Download Center, Intel StrataFlash Memory (J3) datasheet</a>	–	Intel
<a href="#">MultiMedia Card Association specification</a>	–	JEDEC
<a href="#">Simple DirectMedia Layer Cross-platform Development Library</a>	–	Simple DirectMedia Layer
<a href="#">Virtual I/O Device (VIRTIO) Version 1.0</a>	–	OASIS Open